Zagadnienia na obronę 2025 – opracowane

Cybera 2021/2022

Krajowy system cyberbezpieczeństwa....5

Źródła prawa w polskim systemie prawa i ich znaczenie dla regulacji systemu cyberbezpieczeństwa w Polsce....5

Znaczenie incydentów, ich kategorie i etapy obsługi oraz zadania zespołów CSIRT w Krajowym Systemie Cyberbezpieczeństwa....5

Operatorzy usług kluczowych i ich obowiązki w zakresie cyberbezpieczeństwa. 6

Wprowadzenie do sieci Internet....7

Protokoły sieciowe IPv4, IPv6, TCP i UDP oraz DNS, DHCP i NAT....7

Działanie sieci w kontekście warstw modelu OSI/ISO....12

Protokoły rutingu....12

Wstęp do informatyki....14

Budowa i zasady funkcjonowania mikroprocesorów....14

Architektura systemów komputerowych....15

Algorytmy, przetwarzanie danych i ochrona zasobów systemu komputerowego....16

Wprowadzenie do prawa karnego i postępowania karnego....17

Pojęcie prawa karnego, przedmiot odpowiedzialności karnej i przebieg procesu karnego....17

Podstawowe zasady i pojęcia z zakresu prawa karnego i postępowania karnego (przestępstwo, wina, kara, itp.)....19

Czyny zabronione związane ze sferą cyberbezpieczeństwa i odpowiedzialność za nie w kontekście prawa karnego....21

Bezpieczeństwo systemów i sieci teleinformatycznych....22

Kryptograficzne metody ochrony danych....22

Protokoły i usługi bezpieczeństwa oraz ochrona danych poufnych i treści multimedialnych....24

Zagrożenia i detekcja ataków sieciowych....27

Podstawy programowania....28

Język C – cechy języka, zastosowania, kompilacja, sposoby alokacji pamięci i wyciek pamięci, itp.....28

Struktury danych, wskaźniki, funkcje i instrukcje....29

Pojęcie rekurencji i praktyczne przykłady użycia....31

Projektowanie i analiza algorytmów....32

Algorytmy i ich konstrukcja oraz złożoność obliczeniowa....32

Analiza złożonych struktur danych....33

Sortowanie, haszowanie, wyszukiwanie, kodowanie i kompresja....35
Wykrywanie incydentów....37
Pojęcie i znaczenie incydentu oraz klasyfikacja incydentów i przykładowe bazy (MITRE ATT&CK i inne)....37
Sposoby wykrywania i reagowania na incydenty, systemy IDS/IPS i SIEM, jednostki SOC oraz działanie zespołów CSIRT i CERT....38
Etapy "cyber kill chain"....40
Zarządzanie bezpieczeństwem informacji....40
Normy i standardy w zakresie budowy systemu zarządzania bezpieczeństwem informacji....40
Szacowanie ryzyka....41
Identyfikacja i inwentaryzacja zasobów informacyjnych oraz audyt wewnętrzny w organizacji....41
Probabilistyka i statystyka....42
Zmienna losowa, wartość oczekiwana i kwantyle....42
Rozkłady ciągłe i dyskretne (jednostajny, Poissona, wykładniczy, normalny)...44
Średnia, wariancja i przedziały ufności....45
Fizyka....45
Fala elektromagnetyczna jako nośnik informacji....45
Stan kwantowy jako sposób na zapis, transmisję i przetwarzanie informacji kwantowej....47
Podstawy mechaniki kwantowej pozwalające zrozumieć istotę działania komputera kwantowego i kryptografii kwantowej....48
Systemy operacyjne....51
Programy, procesy i wątki w systemie operacyjnym oraz pojęcie stosu i sterty w kontekście pamięci wirtualnej systemu operacyjnego....51
Techniki komunikacji międzyprocesowej....53
Gniazda sieciowe, komunikacja TCP/IP pomiędzy dwoma systemami komputerowymi....54
Programowanie skryptowe....54
Python jako język skryptowy (model obiektowy, paradygmat programowania funkcyjnego, popularne implementacje, itp.)....54
JavaScript, HTML i CSS....55
Realizacja asynchronicznych zapytań HTTP i bezpieczeństwo połączeń HTTP..56
Bezpieczeństwo lokalnych sieci komputerowych....56
Segmentacja sieci w warstwie 2 modelu OSI/ISO....56
Protokoły zapewnienia niezawodności w sieciach Ethernet....57
Cyberbezpieczeństwo w sieciach Ethernet....58
Kryptografia....61

Kryptografia symetryczna i asymetryczna (cechy, wyzwania, stosowane szyfry blokowe, przykładowe algorytmy, itp.) 61
Kryptograficzne funkcje skrótu 62
Wymagania związane z bezpieczeństwem kluczy kryptograficznych i zarządzanie kluczami 63
Analiza malware 64
Typy oprogramowania złośliwego i sposoby działania oraz ich wykrywanie i analiza 64
Inżynieria wsteczna oprogramowania i techniki ukrywania złośliwego oprogramowania (obfuskacja, pakowanie, szyfrowanie, itp.) 65
Informatyka śledcza i wykrywanie anomalii 66
Bazy danych 67
Porównanie relacyjnych baz danych z nierelacyjnymi (NoSQL) 67
Transakcja bazodanowa i jej wymagane cechy 68
Język SQL – cechy języka, problem interpretacji NULL, zagadnienie kolejności wierszy i kolumn w tabeli 69
Bezpieczeństwo bezprzewodowych sieci komputerowych 70
Podstawy transmisji bezprzewodowej oraz metody wielodostępu do kanału radiowego w sieci Wi–Fi 70
Mechanizmy bezpieczeństwa stosowane w sieciach Wi–Fi (np. WPA) i standardzie Bluetooth 72
Podatności i wady mechanizmów WEP i WPS oraz ataki na sieci Wi–Fi 74
Bezpieczeństwo oprogramowania 76
Wybór języka programowania a bezpieczeństwo tworzonej przy jego pomocy aplikacji 76
Bezpieczeństwo aplikacji internetowych i mobilnych – sposoby ochrony, podatności, ataki, ocena bezpieczeństwa 77
Bezpieczne przechowywanie i przetwarzanie danych (przechowywanie haseł, walidacja danych, dobre i złe praktyki programistyczne itp.) 79
Ochrona danych osobowych i technologie wzmacniające prywatność 80
Prywatność użytkowników systemów teleinformatycznych i techniki śledzenia 80
Ochrona i przetwarzanie danych osobowych oraz informacji niejawnych 83
Wycieki danych 85
Bezpieczeństwo w sieciach rozległych 87
Mechanizmy bezpieczeństwa w sieciach rozległych i zabezpieczenie dostępu do urządzeń i usług 87
Bezpieczeństwo poprzez zapory sieciowe 89
Protokoły bezpieczeństwa i sieci VPN 89
Bezpieczeństwo systemów komórkowych i sieci 5G 91
Zjawiska w kanale radiowym charakterystyczne dla radiokomunikacji ruchomej, metody wielodostępu i problematyka ochrony przed PEM 91

Czynniki wpływające na realną przepływność transmisji danych dostępnej dla użytkownika sieci komórkowej i funkcja układów wieloantenowych.....92
Ewolucja systemów 4G – 5G.....94
Szpiegostwo przemysłowe.....99
Szpiegostwo, wywiad i kontrwywiad.....99
Działania socjotechniczne ukierunkowane na manipulację pracowników i nielegalne zdobywanie informacji.....99
Sposoby przeciwdziałania przestępczości szpiegostwa przemysłowego.....100
Testy penetracyjne.....101
Standardy, dobre praktyki i przykłady wykonywania skutecznych testów penetracyjnych.....101
Skanowanie w celu wyszukiwania podatności i automatyzacja testów penetracyjnych.....101
Zabezpieczanie aplikacji i raportowanie wyników pentestów.....102
Programowanie sieciowe wspierające aplikacje bezpieczne.....103
Aplikacje korzystające z protokołu TCP, UDP lub SCTP oraz problematyka ich bezpieczeństwa.....103
Funkcje warstwy sieciowej i warstwy transportowej w programach sieciowych 105
Adresacja typu broadcast i multicast w sieci IP.....106
Bezpieczeństwo infrastruktury krytycznej i systemów sterowania przemysłowego 107
Niezawodność i cyberbezpieczeństwo infrastruktury krytycznej.....107
Techniki analizy zagrożeń i analizy ryzyka.....108
Polityka cyberbezpieczeństwa w przedsiębiorstwie.....109
Praktyczne wykorzystanie informacji z otwartych źródeł.....109
Biały wywiad.....109
Narzędzia OSINT.....110
Pozyskiwanie danych i tworzenie raportów.....110

Krajowy system cyberbezpieczeństwa

Źródła prawa w polskim systemie prawa i ich znaczenie dla regulacji systemu cyberbezpieczeństwa w Polsce

Źródła prawa:

- **Konstytucja** – najwyższy akt prawny, mówi, że jak coś nie powiedziała to resztą się zajmują niżej wymienione
- **Ustawy** – regulują konkretny obszar, np. Ustawa Kodeks Karny, Ustawa o KSC, – Akty uchwalane przez parlament (np. OUK)
- **Ratyfikowane ustawy międzynarodowe** – po przyjęciu przez Polskę mają moc wyższą niż ustawy krajowe (np. ~~NIS~~ NIS nie jest ustawą, tylko dyrektywą)
- **Rozporządzenia** – unijne RODO (Rozporządzenie o Ochronie Danych Osobowych) (ang. GDPR – General Data Protection Regulation), ministrów itd. oraz regulacje techniczne, np. szczegóły dotyczące zgłaszania incydentów bezpieczeństwa.

Znaczenie dla regulacji systemu cyberbezpieczeństwa w Polsce:

- Tworzenie ram prawnych
- Walka z cyberprzestępczością
- Ochrona danych osobowych
- Zapewnienie zgodności z prawem UE

NIS – pierwsze europejskie prawo w zakresie cyberbezpieczeństwa. (EU: 2016, PL: 2018 – Ustawa o KSC).

Wymusza wdrożenie odpowiednich środków technicznych i organizacyjnych, aby zarządzać ryzykiem dla bezpieczeństwa sieci oraz systemów informatycznych. Zawiera przepisy dotyczące zgłaszania incydentów bezpieczeństwa.

NIS jest dyrektywą, dlatego żaden podmiot ani obywatel nie stosuje się do tych przepisów. Są to jedynie swego rodzaju rekomendacje dla państw, które muszą je spełnić. Implementacją NIS w polskim prawie jest KSC.

NIS2 – zwiększa wymogi co do zarzadzania ryzykiem i zgłaszania incydentów.

Ujednolica obowiązki dla dwóch głównych grup, czyli **operatorów usług kluczowych** (np. energia, transport, woda) oraz **dostawców usług cyfrowych** (jeszcze administracja publiczna).

To również jest dyrektywa. Implementacją NIS2 w polskim prawie będzie nowe KSC. Nowe KSC jeszcze w Polsce nie obowiązuje (bo nadal trwają nad nim prace).

ENISA (Europejska Agencja Bezpieczeństwa Sieci i Informacji) – agencja Unii Europejskiej odpowiedzialna za zapewnienie wysokiego i efektywnego poziomu bezpieczeństwa w sieciach i systemach informatycznych w Unii Europejskiej.

Znaczenie incydentów, ich kategorie i etapy obsługi oraz zadania zespołów CSIRT w Krajowym Systemie Cyberbezpieczeństwa

Rodzaje incydentów:

- **Incydent krytyczny** – incydent, skutkujący znaczną szkodą dla bezpieczeństwa lub porządku publicznego, interesów międzynarodowych, interesów gospodarczych, działania instytucji publicznych, praw i wolności obywatelskich lub życia i zdrowia ludzi
- **Incydent poważny** – incydent, który powoduje lub może spowodować poważne obniżenie jakości lub przerwanie ciągłości działania świadczonej usługi kluczowej
- **Incydent istotny** – incydent, który ma istotny wpływ na świadczenie usługi cyfrowej
- **Incydent o podmiocie publicznym** – incydent, który powoduje lub może spowodować obniżenie jakości lub przerwanie realizacji zadania publicznego realizowanego przez podmiot publiczny
- **Incydent zwykły** – zdarzenie, które ma lub może mieć niekorzystny wpływ na cyberbezpieczeństwo – nie podlega zgłoszeniu, ale też należy go obsługiwać

Obsługa incydentu:

Wykrywanie –> Rejestrowanie –> Analizowanie –> Klasyfikowanie –> Priorytetyzowanie –> Podejmowanie działań naprawczych –> Ograniczenie skutków incydentu

Przekazywanie informacji o incydencie przez operatora usługi kluczowej:

- Operator zgłasza incydent niezwłocznie, nie później niż w ciągu 24H od momentu wykrycia, do właściwego CSIRT MON, CSIRT NASK lub CSIRT GOV,
- Operator współdziała podczas obsługi incydentu poważnego i incydentu krytycznego z właściwym CSIRT MON, CSIRT NASK lub CSIRT GOV, przekazując niezbędne dane,
- Operator usuwa wskazane podatności oraz informuje o ich usunięciu organ właściwy.

CSIRT GOV – rządowy, **NASK** – samorządowy, lokalny, obywatelski, **MON** – gospodarczo–obronny

Do wybranych podstawowych zadań CSIRT GOV należy:

- Monitorowanie zagrożeń cyberbezpieczeństwa i incydentów na poziomie krajowym,
- Szacowanie ryzyka związanego z ujawnionym zagrożeniem cyberbezpieczeństwa oraz zaistniałymi incydentami, w tym prowadzenie dynamicznej analizy ryzyka,
- Przekazywanie informacji dotyczących incydentów i ryzyka podmiotom krajowego systemu cyberbezpieczeństwa,
- Wydawanie komunikatów o zidentyfikowanych zagrożeniach cyberbezpieczeństwa,
- Reagowanie na zgłoszone incydenty,

- Prowadzenie i koordynacja systemu wczesnego ostrzegania o zagrożeniach występujących w sieci Internet,
- Przeprowadzanie oceny bezpieczeństwa systemów teleinformatycznych.

Operatorzy usług kluczowych i ich obowiązki w zakresie cyberbezpieczeństwa

Operatorem usługi kluczowej (zwany dalej **OUK**) jest podmiot, posiadający jednostkę organizacyjną na terytorium Rzeczypospolitej Polskiej, wobec którego organ właściwy (Minister odpowiedzialny za dany dział administracji rządowej) wydał decyzję o uznaniu za OUK.

Organ właściwy wydaje decyzję o uznaniu podmiotu za OUK, jeżeli:

- Podmiot świadczy usługę, która ma kluczowe znaczenie dla utrzymania krytycznej działalności społecznej lub gospodarczej, zwaną dalej „**usługą kluczową**”, wymienioną w wykazie usług kluczowych,
- Świadczenie tej usługi kluczowej zależy od systemów informacyjnych,
- Incydent miałby istotny skutek zakłócający dla świadczenia usługi kluczowej przez tego operatora.

Do najważniejszych grup obowiązków OUK należą:

- Zarządzanie ryzykiem (w tym szacowanie ryzyka),
- Wdrożenie odpowiednich i proporcjonalnych do oszacowanego ryzyka środków technicznych i organizacyjnych
    - W tym utrzymanie i bezpieczną eksploatację systemu informacyjnego, bezpieczeństwo fizyczne i środowiskowe, bezpieczeństwo i ciągłość dostaw, wdrażanie, dokumentowanie i utrzymywanie planów działania
- Zbieranie informacji o zagrożeniach cyberbezpieczeństwa i podatnościach na incydenty,
- Zgłaszanie incydentu poważnego do właściwego zespołu CSIRT,
- Obsługa incydentów i współpraca w tym zakresie z właściwym CSIRT,
- Zapewnienie poufności, integralności i dostępności danych,
- Wyznaczenie osoby kontaktowej na potrzeby krajowego systemu cyberbezpieczeństwa.

Usługi kluczowe – sektory:

- energetyczny,
- transportowy,
- bankowy i infrastruktury rynków finansowych,
- ochrony zdrowia,
- zaopatrzenia w wodę pitną (wraz z dystrybucją,
- infrastruktury cyfrowej.

Wprowadzenie do sieci Internet

Główne mechanizmy przełącznika:

1. **Learning:** Uczy się, które urządzenia są podłączone do konkretnych portów.
2. **Flooding:** Zalewa sieć ramkami, gdy nie zna odbiorcy.
3. **Forwarding:** Przesyła ramki tylko do właściwego portu na podstawie tablicy MAC.
4. **Filtering:** Blokuje niepotrzebne lub nieautoryzowane ramki.

Protokoły sieciowe IPv4, IPv6, TCP i UDP oraz DNS, DHCP i NAT

IPv4 – protokół adresów

- 32-bitowa liczba – 4 pola po 8 bitów, w formacie dziesiętnym, oddzielane kropkami
- Przydzielanie adresu: APIPA lub DHCP
- IPsec: nieobowiązkowy
- Wykorzystuje ARP (Address Resolution Protocol) do odnajdywania fizycznego adresu, na przykład adresu MAC lub adresu łącza powiązanego z adresem IPv4
- Wykorzystywane typy adresów:
  - Unicast,
  - Multicast,
  - Broadcast

<table>
<tr><td>Wersja</td><td>IHL</td><td>Typ usługi</td><td colspan="2">Długość pakietu</td></tr>
<tr><td colspan="3">Identyfikacja</td><td>Flaga</td><td>Przesunięcie fragmentu</td></tr>
<tr><td>TTL</td><td colspan="2">Protokół</td><td colspan="2">Suma kontrolna nagłówka</td></tr>
<tr><td colspan="5">Adres źródłowy</td></tr>
<tr><td colspan="5">Adres docelowy</td></tr>
<tr><td colspan="4">Opcje</td><td>Wypełnienie</td></tr>
</table>

- **Wersja** – zawiera numer wersji protokołu IP (w tym wypadku to będzie **IPv4**).
- **Długość nagłówka (IHL)** – określa rozmiar nagłówka pakietu.
- **Typ usługi (ToS)** – zawiera 8-bitową wartość, która używana jest do określenia priorytetu każdego pakietu.
- **Długość pakietu** – to pole podaje w bajtach całkowitą wielkość pakietu, zawierającą nagłówek oraz dane.
- **Identyfikacja** – to pole jest używane do jednoznacznego identyfikowania fragmentów podzielonego pakietu IP.
- **Flaga MF (ang. More Fragments)** – pojedynczy bit używanym z polem przesunięcia fragmentu do podziału i rekonstrukcji pakietów. Gdy bit **flagi MF** jest ustawiony, oznacza to, że dany fragment nie jest ostatnim

fragmentem pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością **MF=1**, sprawdza pole przesunięcia fragmentu, gdzie należy umieścić ten fragment podczas rekonstrukcji pakietu. Kiedy host odbierający zauważy przybywający pakiet z ustawioną wartością **MF=0** i niezerową wartość w polu przesunięcia fragmentu, umieszcza ten fragment jako ostatni kawałek rekonstruowanego pakietu.

- **Flaga DF** (ang. Don't Fragment) – Jeżeli **flaga DF** jest ustawiona, wtedy fragmentacja tego pakietu nie jest dozwolona.
- **Przesunięcie fragmentu** – pole stosowane podczas rekonstrukcji podzielonego przez ruter pakietu. Wskazuje porządek, w jakim ma być ustawiony każdy z pakietów podczas rekonstrukcji.
- **Czas życia (TTL)** – 8 bitowe pole, które określa pozostały czas życia pakietu. Wartość TTL jest zmniejszana o co najmniej 1 za każdym razem, gdy pakiet przechodzi przez ruter (tj. za każdym przeskokiem). Kiedy wartość osiąga 0, ruter porzuca pakiet i jest on usunięty ze strumienia danych w sieci. Mechanizm ten chroni pakiety, które nie mogą osiągnąć celu przed przekazywaniem w nieskończoność pomiędzy ruterami w tzw. pętli rutingu.
- **Protokół** – ta 8-bitowa wartość, określająca wykorzystany protokół warstwy wyższej (transportowej), np. **UDP** lub **TCP**.
- **Suma kontrolna nagłówka** – pole używane jest do sprawdzenia błędów nagłówka pakietu.
- **Docelowy adres IP** – adres IP urządzenia, do którego kierowane są dane.
- **Źródłowy adres IP** – adres IP urządzenia, który jest nadawcą danych.
- **Opcje** – jest to miejsce na dodatkowe pola w nagłówku IPv4 do obsługi innych usług. Jest ono jednak rzadko używane.

Adresy prywatne/lokalne:

- Klasa A - 10.x.x.x (maska 8 bitów)
- Klasa B - 172.16.x.x - 172.31.x.x (maska 12 bitów)
- Klasa C - 192.168.x.x (maska 16 bitów)

IPv6 – protokół adresów, następca IPv4

- **128-bitowa** liczba – 8 pól po 16 bitów, w formacie szesnastkowym, oddzielane dwukropkami
- **Przydzielanie adresu**: urządzenie generuje adres, gdy połączy się z siecią dzięki IRDP i NDP, DHCP też możliwy
- **IPsec**: obowiązkowy
- Funkcje ARP zintegrowane w samym protokole IPv6 – nie istnieje ARP6
- **Wykorzystywane typy adresów**:
    - Unicast,
    - Multicast,
    - Anycast,
    - Lokalne

<table>
<tr><th>Bity</th><th>0-3</th><th>4-7</th><th>8-11</th><th>12-15</th><th>16-19</th><th>20-23</th><th>24-27</th><th>28-31</th></tr>
<tr><td>0</td><td>Wersja</td><td colspan="2">Priorytet</td><td colspan="5">Etykieta przepływu</td></tr>
<tr><td>32</td><td colspan="4">Długość danych</td><td colspan="2">Następny nagłówek</td><td colspan="2">Limit przeskoków</td></tr>
<tr><td>64</td><td colspan="8" rowspan="4">Adres źródłowy (128 bitów)</td></tr>
<tr><td>96</td></tr>
<tr><td>128</td></tr>
<tr><td>160</td></tr>
<tr><td>192</td><td colspan="8" rowspan="4">Adres docelowy (128 bitów)</td></tr>
<tr><td>224</td></tr>
<tr><td>256</td></tr>
<tr><td>288</td></tr>
</table>

- **Wersja** - definiująca wersję protokołu, w przypadku IPv6 pole te zawiera wartość 6.
- **Klasa ruchu** - określa sposób w jaki ma zostać potraktowany pakiet danych. Zastępuje pole **Type of Service** z IPv4.
- **Etykieta przepływu** - pole służące do oznaczania strumienia pakietów IPv6.
- **Długość danych** - wielkość pakietu, nie wliczając długości podstawowego nagłówka,
- **Następny nagłówek** - pole określające typ informacji znajdujących się za podstawowym nagłówkiem. Informacja ta może określać protokół warstwy wyższej TCP/UDP, bądź też może być nowym opcjonalnym nagłówkiem rozszerzeń.
- **Limit przeskoków** - określa ilość węzłów, po odwiedzeniu których pakiet zostaje porzucony. Zastępuje pole **Time to Live** z IPv4.
- **Adres źródłowy** - adres węzła, który wysłał pakiet.
- **Adres docelowy** - adres węzła, do którego adresowany jest pakiet.

UDP (User Datagram Protocol) – bezpołączeniowy protokół datagramowy, brak mechanizmów kontroli przepływu i retransmisji. Jest protokołem zawodnym, tzn. nie chroni przed:

- zmianą kolejności pakietów,
- gubieniem pakietów,
- duplikacją pakietów.

Siłą tego protokołu jest jego prostota. Znajduje się nad protokołem IP, który też jest bezpołączeniowy, dlatego sam nie musi wiele robić.

Przykłady: transmisja dźwięku i video, DHCP, TFTP, DNS

Port źródłowy	Port docelowy
Długość	Suma kontrolna
Dane	

TCP (Transmission Control Protocol) – połączeniowy, niezawodny, mechanizmy kontroli przepływu (zmiana rozmiaru okna) i retransmisji. Odpowiada za:

- nawiązanie i zakończenie połączenia transportowego,
- zapewnienie niezawodności przekazu danych,
- potwierdzenia,
- retransmisje,
- sprawdzanie sum kontrolnych,
- odtwarzanie kolejności danych,
- kasowanie duplikatów,
- sterowanie przepływem (ang. flow control).

Przykłady: transfer plików, HTTP, FTP, e–mail

Port źródłowy			Port docelowy
Numer sekwencyjny			
Numer potwierdzenia			
Przesunięcie	Zarezerwowane	Flagi TCP	Szerokość okna
Suma kontrolna			Wskaźnik priorytetu
Opcje	Uzupełnienie		
Dane			

DNS (Domain Name System) – hierarchiczny rozproszony system nazw domen zajmujący się tłuczeniem nazw mnemonicznych jak google.com na adresy IP, czyli x.x.x.x.

Serwery wyżej w hierarchii przechowują informacje o domenach najwyższych poziomów. Serwery w hierarchii mogą wskazywać między sobą, prowadząc pytającego do serwera posiadającego zapis tłumaczący daną nazwę na adres.

Adresy stron są trzymane na serwerze DNS za pomocą rekordów:

- A Record – IPv4
- AAAA Record – IPv6

DHCP (Dynamic Host Configuration Protocol) – protokół zajmujący się dynamiczną konfiguracją hostów, czyli nadawaniem im podstawowych parametrów w sieci zależnie od ustawień

- **DHCPDISCOVER** – klient DHCP wysyła ten komunikat w trybie *broadcast,* aby znaleźć serwer DHCP,
- **DHCPOFFER** – serwery DHCP odpowiadają tym komunikatem na komunikat DHCP, oferując dzierżawę adresu IP klientowi,

- **DHCPREQUEST** — klient akceptuje pierwszą ofertę wysłaną przez DHCPOFFER, która oferowała dzierżawę adresu IP, żąda tego adresu,
- **DHCPACK** — jeżeli adres, którego żąda klient może być użyty, to serwer akceptuje to żądanie wysyłając właśnie ten komunikat

NAT (Network (czasami Native) Address Translation) – technika przesyłania ruchu sieciowego poprzez router, która wiąże się ze zmianą źródłowych lub docelowych adresów IP, zwykle również numerów portów TCP/UDP pakietów IP podczas ich przepływu. Zmieniane są także sumy kontrolne (zarówno w pakiecie IP, jak i w segmencie TCP/UDP), aby potwierdzić wprowadzone zmiany.

Aby działać potrzebuje utrzymywać tablicę translacji adresów (i numerów portów), dokonywać translacji adresów (i numerów portów), przeliczać odpowiednie sumy kontrolne.

- Typy adresów: Inside/Outside – Local/Global
- Najczęściej między siecią prywatną a publiczną –> wiele urządzeń w sieci prywatnej ma ten sam adres publiczny.

Działanie sieci w kontekście warstw modelu OSI/ISO

Model **OSI/ISO** składa się z warstw, które mają reprezentować poszczególne etapy/poziomy komunikacji. Im wyższa warstwa, tym bardziej skomplikowany format informacji i na odwrót. Idąc w górę, dane są enkapsulowane. Warstwy:

- **7 – aplikacji – DANE**, np. protokół SNMP, HTTP, Polaczenie procesów sieciowych z aplikacjami zapewnienie usług sieciowych procesom aplikacji (takich jak np. poczta elektroniczna, www)
- **6 – prezentacji – DANE**, reprezentacja danych i informacji; formaty JPEG/ASCII/GIF itd.; mało istotne w zakresie sieci, zarządzanie sposobami kodowania danych, szyfrowanie danych
- **5 – sesji – DANE**, ustanawianie i zamykanie sesji pomiędzy aplikacjami, zarządzanie sesjami
- **4 – transportowa – SEGMENT**, np. TCP, UDP, zajmuje się połączeniami, kontrola transmisji, niezawodność transmisji, detekcja błędów, porządkowanie pakietów, retransmisje
- **3 – sieci – PAKIET**, np. IP, ICMP, dotyczy połączeń w ramach sieci, adresacja IPv4 / IPv6, brak detekcji i korekcji błędów
- **2 – łącza danych – RAMKA**, np. ARP, dotyczy połączeń pomiędzy urządzeniami, adresy MAC, prawidłowość transmitowanych danych;
- **1 – fizyczna – BIT** (ponieważ sygnał może tylko być lub nie być, 1/0); fizyczne połączenie i transmisja danych w łączu miedzianym / światłowodowym / radiowym itp.

Model **TCP/IP** upraszcza model **OSI/ISO** na potrzeby pracy z sieciami:

- **4 – aplikacji** – połączenie warstw 5+6+7 OSI/ISO
- **3 – transportowa** – warstwa 4 (transportowa)
- **2 – sieci** – warstwa 3 (sieci)
- **1 – dostępu do sieci** – warstwy 1+2

Korzyści ze stosowania modelu OSI:

- Mniejsza złożoność,

- Standaryzacja interfejsów,
- Projektowanie modułowe,
- Współdziałanie technologii,
- Przyspieszony rozwój,
- Uproszczenie procesu nauczania.

Protokoły rutingu

Protokoły rutingu (ang. **routing protocols**) to zestaw reguł i mechanizmów, które umożliwiają routerom komunikację i wymianę informacji na temat struktury sieci. Dzięki nim routery mogą podejmować decyzje o trasach, którymi przesyłane są dane w sieciach komputerowych.

Statyczny – wpisanie na twardo reguły rutingu do tabeli:

- Zalety:
  - Może służyć jako rozwiązanie zapasowe (jako trasa alternatywna)
  - Łatwość konfigurowania
  - Nie zajmuje zasobów rutera (np. CPU i pamięci)
  - Wymaga znajomości topologii sieci –> bezpieczeństwo
- Wady:
  - Każda zmiana w sieci wymaga ręcznej rekonfiguracji
  - Kłopoty związane ze skalowalnością w rozległych topologiach

Dynamiczny – odpowiedni protokół ustawia trasy:

- Elementy:
  - **Protokół komunikacyjny**: Zbiór komunikatów służących wykrywaniu, nawiązywaniu sąsiedztwa oraz do wymiany informacji o trasach
  - **Algorytm doboru trasy**: Jest wykorzystywany do wyznaczania najlepszej trasy prowadzącej do określonej sieci docelowej
- Zalety:
  - Zmiany w sieci są automatycznie zauważane
  - Topologia sieci może się zmieniać
  - Zysk czasu podczas konfiguracji dla dużych sieci
- Wady:
  - Zajmuje zasoby ruterów i sieci
  - Nieodpowiednio skonfigurowany może skutecznie zablokować komunikację w sieci

Protokoły wektora odległości:

- Ruter przekazuje okresowo do wszystkich swoich sąsiadów (tylko sąsiadów) kopie swojej tablicy rutingu wraz z metrykami. Jest to informacja o adresach dostępnych z danego routera bezpośrednio oraz za pośrednictwem innych routerów.
- Trasy są rozgłaszane jako wektory odległości i kierunku,
- Niekompletna wizja topologii sieci,
- Aktualizacje tablic rutingu następują w przypadku zmian topologii sieci,
- Ogólne, periodyczne uaktualnienia,
- Ograniczona do małych i średnich sieci.

Protokoły stanu łącza:

- Ruter przekazuje okresowo do wszystkich routerów w sieci (nie tylko sąsiadów) już nie całą kopię tablicy rutingu, a jedynie informację o podsieciach bezpośrednio do niego przyłączonych wraz z ich metrykami
- Pełna wizja topologii sieci,
- Topologia jest uaktualniania na bieżąco,
- Uaktualnienia nie są periodyczne,
- Dobrze skalowalne w dużych sieciach, szybsza i bardziej stabilna.

RIP (Routing Information Protocol):

- Protokół oparty na algorytmie **wektora odległości**,
- Metryka służąca do wyboru ścieżki to liczba przeskoków (hop count),
- Maksymalna liczba przeskoków: 15 (16 oznacza trasę niedostępną),
- Używa odświeżania co 30 sekund.

OSPF (Open Shortest Path First):

- Protokół oparty na algorytmie **stanu łącza** (link–state),
- Działa na zasadzie hierarchii obszarów (areas),
- Szybki i wydajny, szczególnie w dużych sieciach.

BGP (Border Gateway Protocol):

1) Oparty na algorytmie **wektora ścieżki**
    - Algorytm wektora ścieżki przechowuje i wymienia informacje nie tylko o odległości do celu, ale także o całej ścieżce, jaką pakiety muszą pokonać, aby dotrzeć do danego miejsca,
2) Podstawowy protokół rutingu w Internecie (między operatorami Internetowymi),
3) Wymienia informacje między systemami autonomicznymi (AS).

Metryka — wartość wykorzystywana przez protokół rutingu do wyznaczenia, która z dostępnych tras do sieci docelowej jest lepsza niż inne. Rodzaje:

- przepływność łącza (bandwidth),
- koszt (cost),
- opóźnienie (delay),
- liczba skoków (hop count),
- obciążenie łącza (load),
- niezawodność (reliability),

Wstęp do informatyki

Budowa i zasady funkcjonowania mikroprocesorów

W prawie każdym mikroprocesorze możemy wyróżnić następujące bloki:

- **ALU – jednostka arytmetyczno-logiczna (Arithmetic Logic Unit)**, wykonuje ona operacje logiczne na dostarczonych jej danych, podstawowy

zestaw to: dodawanie, podstawowe operacje logiczne (AND, XOR, OR, NOT), oraz przesunięcia bitowe w lewo i w prawo. W bardziej złożonych mikroprocesorach zestaw ten jest znacznie bogatszy.

- **CU – układ sterowania (Control Unit)**, zwany też dekoderem rozkazów. Odpowiedzialny jest on za dekodowanie dostarczonych mikroprocesorowi instrukcji i odpowiednie sterowanie pozostałymi jego blokami
    - Na przykład, jeśli zdekodowaną instrukcją będzie dodawanie, CU odpowiednio ustawi sygnały sterujące, by ALU wykonała tę właśnie operację
- **Rejestry** – umieszczone wewnątrz mikroprocesora komórki pamięci o niewielkich rozmiarach (najczęściej 4/8/16/32/64/128 bitów) służące do przechowywania tymczasowych wyników obliczeń (rejestry danych) oraz adresów lokacji w pamięci operacyjnej (rejestry adresowe). Proste mikroprocesory mają tylko jeden rejestr danych zwany akumulatorem. Oprócz rejestrów danych i rejestrów adresowych występuje też pewna liczba rejestrów o specjalnym przeznaczeniu:
    - **PC – licznik rozkazów (Program Counter)** – zawiera on adres komórki pamięci zawierającej następny rozkaz do wykonania
    - **IR – rejestr instrukcji (Instruction Register)** – zawiera on kod aktualnie wykonywanej przez procesor instrukcji.
    - **SP – wskaźnik stosu (Stack Pointer)** – zawiera adres wierzchołka stosu

Mikroprocesor komunikuje się z otoczeniem za pomocą **szyny danych** i **szyny adresowej**.

Generalnie każdy bardziej skomplikowany mikroprocesor można zaklasyfikować do jednej z trzech architektur:

- CISC (Complex Instruction Set Computers)
- RISC (Reduced Instruction Set Computers)
- VLIW (Very Long Instruction Word)

Architektura systemów komputerowych

Architektura von Neumanna:

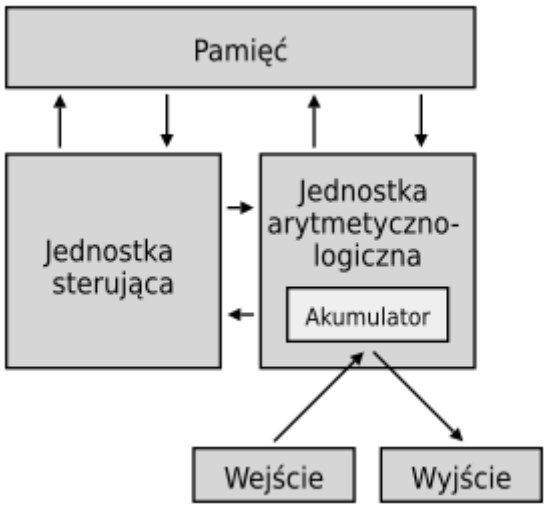
W architekturze tej komputer składa się z czterech głównych komponentów:

- pamięci komputerowej przechowującej dane programu oraz instrukcje programu; każda komórka pamięci ma unikatowy identyfikator nazywany jej adresem
- jednostki sterującej odpowiedzialnej za pobieranie danych i instrukcji z pamięci oraz ich sekwencyjne przetwarzanie
- jednostki arytmetyczno-logicznej odpowiedzialnej za wykonywanie podstawowych operacji arytmetycznych.
- urządzeń wejścia/wyjścia służących do interakcji z operatorem

Jednostka sterująca wraz z jednostką arytmetyczno-logiczną tworzą procesor.

Wszystkie elementy komunikują się ze sobą za pomocą magistrali systemowej (BUS).

Większość komputerów posiada architekturę bazującą na architekturze von Neumanna.



Architektura Harvardzka:

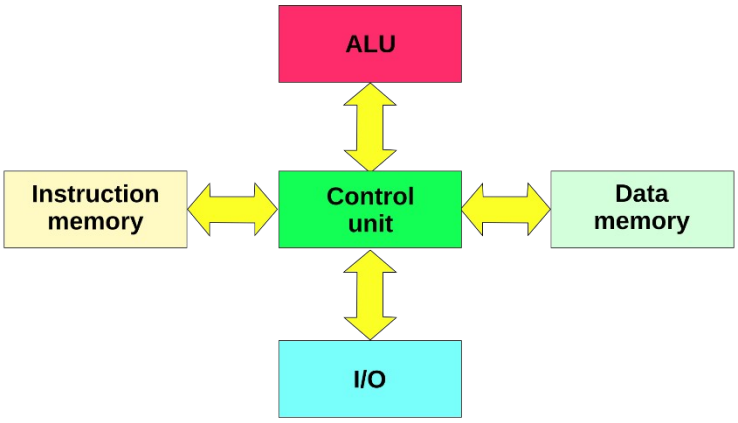
W odróżnieniu od architektury von Neumanna, pamięć danych programu jest oddzielona od pamięci rozkazów.

Podstawowa architektura komputerów zerowej generacji i początkowa komputerów pierwszej generacji.

Prostsza (w stosunku do architektury von Neumanna) budowa przekłada się na większą szybkość działania, ten typ architektury jest wykorzystywany w procesorach sygnałowych oraz przy dostępie procesora do pamięci cache.

Oddzielenie pamięci danych od pamięci rozkazów sprawia, że architektura harwardzka jest stosowana w mikrokomputerach jednoukładowych, w których dane programu są zapisane w nieulotnej pamięci ROM (EPROM/EEPROM), natomiast dla danych tymczasowych wykorzystana jest pamięć RAM (wewnętrzna lub zewnętrzna).

Wszystkie elementy komunikują się ze sobą za pomocą CPU.



Algorytmy, przetwarzanie danych i ochrona zasobów systemu komputerowego

Algorytmy

Algorytmy to precyzyjnie zdefiniowane kroki, które prowadzą do rozwiązania konkretnego problemu. W kontekście systemów komputerowych algorytmy odgrywają kluczową rolę w:

- Sortowaniu i wyszukiwaniu danych (np. algorytmy QuickSort, MergeSort, czy wyszukiwanie binarne).
- Przetwarzaniu równoległym i optymalizacji procesów.
- Uczeniu maszynowym i sztucznej inteligencji, gdzie algorytmy uczące (np. sieci neuronowe, algorytmy gradientowe) są podstawą.
- Algorytmach kryptograficznych, które zabezpieczają dane (np. RSA, AES).

Cechy dobrego algorytmu:

- Efektywność czasowa i przestrzenna.
- Czytelność i prostota implementacji.
- Skalowalność.

Przetwarzanie danych

Przetwarzanie danych obejmuje operacje takie jak zbieranie, przechowywanie, analiza, interpretacja i prezentacja danych. Proces ten można podzielić na kilka kluczowych etapów:

- **Gromadzenie danych**: Zbieranie informacji z różnych źródeł (np. sensory IoT, bazy danych, API).
- **Przechowywanie danych**: Użycie odpowiednich systemów baz danych, takich jak SQL, NoSQL czy rozproszone systemy pamięci (np. Hadoop, BigQuery).
- **Analiza danych**: Techniki analityczne, takie jak analiza statystyczna, eksploracja danych (data mining) i uczenie maszynowe.
- **Wizualizacja danych**: Prezentowanie wyników w sposób przystępny, np. w formie wykresów czy raportów.

Ochrona zasobów systemu komputerowego

Ochrona zasobów systemu komputerowego polega na zapewnieniu integralności, poufności i dostępności danych oraz systemów. Wymaga zastosowania różnych technologii i strategii:

- **Kryptografia**: Szyfrowanie danych w celu zabezpieczenia ich przed nieuprawnionym dostępem.
- **Zapory sieciowe (firewalle)** i **systemy IDS/IPS**: Chronią przed nieautoryzowanym dostępem i monitorują ruch sieciowy.
- **Systemy uwierzytelniania**: Techniki takie jak uwierzytelnianie dwuskładnikowe (2FA) czy użycie biometrii.
- **Kopie zapasowe i odtwarzanie awaryjne**: Zabezpieczanie danych przed ich utratą w wyniku awarii.
- **Aktualizacje oprogramowania**: Regularne aktualizowanie systemów w celu załatania luk bezpieczeństwa.

- **Zarządzanie dostępem**: Ograniczenie dostępu do zasobów tylko dla uprawnionych użytkowników.

aaWprowadzenie do prawa karnego i postępowania karnego

Pojęcie prawa karnego, przedmiot odpowiedzialności karnej i przebieg procesu karnego

Prawo karne – gałąź prawa regulująca zasady odpowiedzialności za czyny zabronione przez prawo, które stanowią przestępstwa.

Jego celem jest ochrona wartości społecznych, takich jak życie, zdrowie, wolność, własność, bezpieczeństwo publiczne, a także zapewnienie porządku prawnego poprzez prewencję i represję wobec sprawców przestępstw.

Element systemu prawnego • Zaliczane do prawa publicznego • Odrębna gałąź prawa • Istnieje w nim relacja podporządkowania • Specyficzne środki prawnej ochrony • Chroni wyłącznie najcenniejsze dobra społeczne

Podział (umowny): prawo karne materialne, procesowe, wykonawcze,

Prawo karne materialne realizuje się wyłącznie poprzez proces karny.

Funkcje prawa karnego:

- **Gwarancyjna**: normy prawa karnego procesowego:
  - Zapewniają ochronę praw człowieka oraz konstytucyjnych praw i wolności uczestników procesu karnego przed arbitralną, to jest nadmierną i bezprawną ingerencją organów procesowych w te prawa,
  - Wyznaczają granice dopuszczalnego działania organów procesowych ingerujących w prawa człowieka, np. tymczasowe zatrzymanie tylko jeśli w KPK jest jasna przesłanka
- **Sprawiedliwościowa**: wynika z potrzeby odpłacenia za krzywdę.
  - Zapewnia sprawiedliwość poprzez ukaranie sprawców czynów zabronionych. Dzięki temu dochodzi do rozładowania negatywnego napięcia społecznego wywołanego przestępstwem.
- **Ochronna**: ustanowienie zakazu zachowań, które zagrażałyby ważnym dla danego społeczeństwa wartościom.
  - Wśród najważniejszych wartości zwykle chronionych przez prawo karne wymienić można życie, zdrowie, wolność czy własność.
- **Kompensacyjna**: Idea, aby sprawca przestępstwa po jego popełnieniu podjął próbę naprawienia wyrządzonej przestępstwem szkody lub krzywdy i niwelował wywołany swoim przestępnym zachowaniem konflikt z pokrzywdzonym (związana z funkcją sprawiedliwościową).
  - Kładzie więc nacisk na aspekt likwidowania przez sprawcę następstw przestępstwa

Przedmiot procesu karnego to **kwestia odpowiedzialności karnej** oskarżonego za zarzucony mu czyn, a nie sama wina – inaczej nie dałoby się wyciągnąć konsekwencji prawnych.

Podstawowe źródło prawa karnego to **ustawa karna** – obecnie **kodeks karny** składający się z 3 części: ogólnej, szczególnej oraz wojskowej

- Orzeczenia sądów nie są źródłem prawa karnego

Prawo karne procesowe to zespół norm prawnych regulujących proces karny, uprawnienia i obowiązki jego uczestników, a także tryb i formy dokonywania czynności procesowych. Obowiązuje:

- w miejscach (całe terytorium Polski),
- w czasie (nie wprowadzono żadnej stałej reguły, która jednoznacznie wskazywałaby na sposób postępowania w razie zmiany prawa w toku trwającego procesu.)

Przedmiot odpowiedzialności karnej

Podstawą faktyczną odpowiedzialności karnej jest **czyn (przestępstwo)** zarzucony oskarżonemu

Podstawą normatywną odpowiedzialności karnej jest przyjęta przez organ procesowy **kwalifikacja prawna tego czynu**.

Sąd nie jest związany kwalifikacją prawną, która może się zmienić, o ile sąd nie wykracza poza granice oskarżenia.

Odpowiedzialności karnej **podlega** ten tylko, kto **popełnia czyn zabroniony pod groźbą kary** przez ustawę obowiązującą w czasie jego popełnienia.

Nie stanowi przestępstwa czyn zabroniony, którego społeczna **szkodliwość jest znikoma**.

Nie popełnia przestępstwa sprawca czynu zabronionego, jeżeli **nie można mu przypisać winy w czasie czynu**.

Przebieg procesu karnego – postępowanie przygotowawcze –> sądowe I instancji (główne) –> sądowe II instancji (odwoławcze)

- **Przygotowawcze**: zbieranie dowodów, stawianie zarzutu (śledztwo lub dochodzenie) przez organy ścigania
- **Postępowanie sądowe**: uniewinnienie, (warunkowe) umorzenie postępowania, skazanie (sąd rejonowy –> okręgowy –> apelacyjny)
- **Postępowanie odwoławcze**: apelacja, zażalenie i sprzeciw, kasacja i wznowienie postępowania
- **Postępowanie wykonawcze** (ale to jest poza procesowym prawem!!) – ma funkcje egzekucyjną i likwidacyjną. Funkcja egzekucyjna polega na wykonaniu prawomocnych wyroków sądu. Funkcja likwidacyjna polega na prawnym usuwaniu skutków skazania lub łagodzeniu jego efektów.

Organem prowadzącym postępowanie przygotowawcze nie jest prokuratura, lecz **prokurator**. Policja może być pomocnicza.

Proces karny: prawnie uregulowana działalność zmierzająca do wykrycia i ustalenia przestępstwa oraz jego sprawcy, osądzenia go za ten czyn oraz wymierzenia mu kar i środków karnych przewidzianych przez prawo materialne.

Modelowy proces karny prowadzony w trybie publicznoskargowym składa się z **trzech stadiów** (ale nie zawsze!). Jeśli przygotowawcze nie da podstaw do wniesienia aktu oskarżenia – wtedy jeden etap.

Jeśli żadna ze stron nie zaskarży wyroku z pierwszej instancji – wtedy dwa etapy.

Strony procesowe w postępowaniu przygotowawczym:

- Czynna →pokrzywdzony || Bierna → podejrzany

Strony procesowe w postępowaniu jurysdykcyjnym:

- Czynne → oskarżyciele || Bierna → oskarżony

Podstawowe zasady i pojęcia z zakresu prawa karnego i postępowania karnego (przestępstwo, wina, kara, itp.)

Przestępstwo – konkretne zachowanie, będące czynem zabronionym pod groźbą określonej kary, czynem bezprawnym, karygodnym i zawinionym. To czyn zabroniony, który zagrożony jest karą:

- grzywny,
- ograniczenia wolności,
- pozbawienia wolności,
- 25 lat pozbawienia wolności,
- dożywotniego pozbawienia wolności.

Formami czynu zabronionego pod groźbą kary są jego:

1) przygotowanie,
2) usiłowanie,
3) dokonanie.

Wykroczenie – konkretne zachowanie, będące czynem zabronionym... (jak wyżej) Nie jest ono przestępstwem z uwagi na zakres kary grożącej za jego popełnienie. To czyn zabroniony, który jest zagrożony karą:

- aresztu (najkrócej pięć dni, a najdłużej 30 dni),
- ograniczenia wolności,
- grzywny do 5000 zł,
- nagany.

Wina – doszukiwanie się istoty winy w elemencie negatywnej (ujemnej) oceny sprawcy i jego zachowania się. Ale nie sposób przypisać człowiekowi winę w oderwaniu od globalnej oceny jego zachowania się uwzględniającej wszystkie okoliczności, w jakich przyszło mu popełnić bezprawny czyn zabroniony pod groźbą kary.

Kara (kryminalna) – konsekwencja popełnienia przestępstwa, którą jest określona przez prawo karne dolegliwość i w której wyraża się dezaprobata czynu i osoby sprawcy

- kara grzywny,
- kara ograniczenia wolności (praca na cele społeczne),
- kara pozbawienia wolności,
- kara 25 lat pozbawienia wolności,
- kara dożywotniego pozbawienia wolności.

Grzywna – jest w systemie kar określonych w kodeksie karnym karą najłagodniejszego rodzaju.

Środek karny – dodatkowa sankcja, np. zakaz prowadzenia pojazdów, pozbawienie praw publicznych, zakaz zajmowania określonego stanowiska.

Zbrodnia – czyn zabroniony zagrożony karą pozbawienia wolności na czas nie krótszy od lat 3 albo karą surowszą.

Występek – czyn zabroniony zagrożony grzywną powyżej 30 stawek dziennych albo powyżej 5000 złotych, albo karą ograniczenia wolności przekraczającą miesiąc

Zasada nullum crimen sine lege – brak przestępstwa bez ustawy, czyli odpowiedzialność karna tylko za czyny określone jako przestępstwa w obowiązującym prawie. (nie ma przestępstwa bez złamania prawa)

Zasada nulla poena sine lege – brak kary bez podstawy w ustawie, kary muszą być jasno określone przez prawo.

Zasada lex retro non agit – prawo nie działa wstecz

te 3 ^ zasady są do funkcji gwarancyjnej!

Zasada domniemania niewinności – oskarżony jest niewinny, dopóki jego wina nie zostanie udowodniona w prawomocnym wyroku.

Czyny zabronione związane ze sferą cyberbezpieczeństwa i odpowiedzialność za nie w kontekście prawa karnego

Art. 267. [Bezprawne uzyskanie informacji]

§ 1. Kto bez uprawnienia uzyskuje dostęp do informacji dla niego nieprzeznaczonej, otwierając zamknięte pismo, podłączając się do sieci telekomunikacyjnej lub przełamując albo omijając elektroniczne, magnetyczne, informatyczne lub inne szczególne jej zabezpieczenie,

podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.

§ 2. Tej samej karze podlega, kto bez uprawnienia uzyskuje dostęp do całości lub części systemu informatycznego.

§ 3. Tej samej karze podlega, kto w celu uzyskania informacji, do której nie jest uprawniony, zakłada lub posługuje się urządzeniem podsłuchowym, wizualnym albo innym urządzeniem lub oprogramowaniem.

Art. 268a. [Niszczenie danych informatycznych]

§ 1. Kto, nie będąc do tego uprawnionym, niszczy, uszkadza, usuwa, zmienia lub utrudnia dostęp do danych informatycznych albo w istotnym stopniu zakłóca lub uniemożliwia automatyczne przetwarzanie, gromadzenie lub przekazywanie takich danych,

podlega karze pozbawienia wolności do lat 3.

§ 2. Kto, dopuszczając się czynu określonego w § 1, wyrządza znaczną szkodę majątkową,

podlega karze pozbawienia wolności od 3 miesięcy do lat 5.

§ 3. Ściganie przestępstwa określonego w § 1 lub 2 następuje na wniosek pokrzywdzonego.

Art. 269. [Uszkodzenie danych informatycznych]

§ 1. Kto niszczy, uszkadza, usuwa lub zmienia dane informatyczne o szczególnym znaczeniu dla obronności kraju, bezpieczeństwa w komunikacji, funkcjonowania administracji rządowej, innego organu państwowego lub instytucji państwowej albo samorządu terytorialnego albo zakłóca lub uniemożliwia automatyczne przetwarzanie, gromadzenie lub przekazywanie takich danych,

podlega karze pozbawienia wolności od 6 miesięcy do lat 8.

§ 2. Tej samej karze podlega, kto dopuszcza się czynu określonego w § 1, niszcząc albo wymieniając informatyczny nośnik danych lub niszcząc albo uszkadzając urządzenie służące do automatycznego przetwarzania, gromadzenia lub przekazywania danych informatycznych.

Art. 269a. [Zakłócenie systemu komputerowego]

Kto, nie będąc do tego uprawnionym, przez transmisję, zniszczenie, usunięcie, uszkodzenie, utrudnienie dostępu lub zmianę danych informatycznych, w istotnym stopniu zakłóca pracę systemu informatycznego, systemu teleinformatycznego lub sieci teleinformatycznej,

podlega karze pozbawienia wolności od 3 miesięcy do lat 5.

Art. 269b. [Wytwarzanie programów komputerowych]

§ 1. Kto wytwarza, pozyskuje, zbywa lub udostępnia innym osobom urządzenia lub programy komputerowe przystosowane do popełnienia przestępstwa:

- Sprowadzenie stanów powszechnie niebezpiecznych dla życia lub zdrowia,
    - Chodzi o automatyczne przetwarzanie, gromadzenie i przekazywanie danych informatycznych
- Przestępstwa przeciwko ochronie informacji (te tutaj podane)
- Przestępstwa przeciwko wiarygodności dokumentów (fałszerstwo faktur lub materialne)

a także hasła komputerowe, kody dostępu lub inne dane umożliwiające nieuprawniony dostęp do informacji przechowywanych w systemie informatycznym, systemie teleinformatycznym lub sieci teleinformatycznej,

podlega karze pozbawienia wolności od 3 miesięcy do lat 5.

Art. 269c. [Kontratyp działania w celu wykrycia błędów w zabezpieczeniach systemów informatycznych]

Nie podlega karze za przestępstwo określone w art. 267 § 2 lub art. 269a, kto działa wyłącznie w celu zabezpieczenia systemu informatycznego, systemu teleinformatycznego lub sieci teleinformatycznej albo opracowania metody takiego zabezpieczenia i niezwłocznie powiadomił dysponenta tego systemu lub sieci o ujawnionych zagrożeniach, a jego działanie nie naruszyło interesu publicznego lub prywatnego i nie wyrządziło szkody.

Bezpieczeństwo systemów i sieci teleinformatycznych

Kryptograficzne metody ochrony danych

1) Steganografia

Ukrywanie przesyłanych informacji tak aby osoba postronna nie była świadoma ich istnienia

- Historycznie np. wydziaranie głowy niewolnikowi, a obecnie LSB (Least Significant Bit) w plikach albo nieużywane pola w protokołach komunikacji

2) Szyfrowanie

Szyfry podstawieniowe

- **proste** (monoalfabetyczne) np. szyfr Cezara, ROT13
- **homofoniczne** – jedna litera jest reprezentowana przez kilka różnych znaków
- **poligramowe** np. szyfr Playfaira (szyfrowane są grupy znaków)
- **wieloalfabetowe** – złożenie szyfrów podstawieniowych, np. szyfr Vigenere'a, szyfr Vernama, One–time pad (ideał)

Szyfry przestawieniowe

- np. szyfr zygzakowaty, transpozycje macierzy z wpisanym tekstem

Szyfry podstawieniowo–przestawieniowe

- np. Enigma
- AES
- DES

Szyfry symetryczne

Szybkie szyfrowanie, problem z bezpieczną dystrybucją/ustaleniem klucza kryptograficznego:

- **Strumieniowe** – Dane wejściowe są szyfrowane (najczęściej XOR-owane) po kolei za pomocą strumienia (keystream) wygenerowanego z klucza
    - ChaCha20 (256 bitów klucza, 92 bitów nonce, 32 bitów countera):

        - Wydajniejszy niż AES (bez przyśpieszenia hardware'owego), przez co jest jego zamiennikiem dla systemów wbudowanych i innych słabszych urządzeń
- **Blokowe** – Jednocześnie szyfrujemy cały blok bitów
    - DES – 64–bitowe bloki
    - 3DES
    - AES (wejście i wyjście: 128 bitów, możliwe długości klucza: 128, 192 lub 256 bitów)
        - Tryby szyfrowania (ECB, CBC, CFB, OFB, CTR)
        - Tryby szyfrowania różnią się:
            - Poziomem bezpieczeństwa
            - Propagacją błędów
            - Wydajnością

Szyfry asymetryczne

Bezpieczne, ale mało wydajne, dlatego są używane np. w TLS do przesyłania klucza dla szyfru symetrycznego

- **RSA** (potęgowanie, arytmetyka modulo, liczby pierwsze; bezpieczeństwo oparte na trudności rozkładu dużych liczb na czynniki pierwsze (faktoryzacji)
- **ElGamal** (bezpieczeństwo oparte na problemie z obliczeniem logarytmu dyskretnego)
- **Krzywe eliptyczne**

3) Wybrane własności bezpiecznych szyfrów:

- **Zrównoważenie** — szyfrogram powinien zawierać tyle samo zer co jedynek (żadna wartość nie jest faworyzowana)
- **Lawinowość** — niewielkie zmiany na wejściu skutkują dużymi zmianami na wyjściu (np. kryterium SAC zakłada, że zmiana pojedynczego bitu wejściowego powoduje zmianę każdego bitu wyjściowego z prawdopodobieństwem ½
- **Nieliniowość** — jest określona jako liczba bitów, które muszą zostać zmienione w tablicy prawdy funkcji boolowskiej, aby osiągnąć najbliższą funkcję afiniczną. Funkcja o wysokiej nieliniowości jest odporna na kryptoanalizę liniową
- **Zupełność** — własność ta jest spełniona, jeśli każdy bit wyjściowy zależy od każdego bitu wejściowego.
- **Niezależność międzybitowa** — zmiana pojedynczego bitu na wejściu, skutkuje zmianami bitów na wyjściu, które są od siebie niezależne.
- **Odpowiedni rozkład wartości profilu XOR w S–bloku** (ochrona przed kryptoanalizą różnicową).
- **Brak korelacj**i, **pseudolosowy ciąg bitów**…

4) Podpis elektroniczny

Szyfrujemy wiadomość lub jej skrót kluczem prywatnym

- Dystrybucja klucza publicznego
    - Samodzielnie
    - Certyfikaty, np. PGP X.509

    - PKI (Infrastruktura klucza publicznego) (używana np. w obrębie firmy)

Podstawowe własności podpisu elektronicznego:

- **Unikalność** – każdy dokument elektroniczny posiada unikalny podpis cyfrowy ściśle związany z dokumentem
- **Integralność** – jakakolwiek zmiana w treści dokumentu podpisanego cyfrowo unieważnia podpis
- **Niezaprzeczalność** – tylko osoba posiadająca klucz prywatny może wygenerować podpis pod dokumentem

Urządzenia do składania podpisu elektronicznego:

- Karta kryptograficza (nośnik certyfikatu)
- Czytnik kart kryptograficznych
- Komputer z oprogramowaniem klienckim

5) Funkcje skrótu

- np. zapisywanie haseł w systemach operacyjnych (np. etc/shadow, więcej – patrz powiązane pytania)
- Kryptograficznymi funkcjami skrótu są, np.:
    - SHA-1
    - SHA-256
    - SHA-3 (Keccak)
    - MD-5 - uważany za niebezpieczny

6) Kryptografia kwantowa (kwantowa dystrybucja klucza)

Kolejna metoda do ustalania kluczy między nadawcą a odbiorcą (uważana za najbezpieczniejszą) np. protokół BB84

- **Polaryzacja fotonów** jako sposób na ustalanie klucza
- **Bezpieczeństwo oparte na prawach fizyki kwantowej**: pomiar zaburza stan kwantowy, oraz to, że nie można sklonować nieznanego stanu kwantowego
- **Aktualny problem do rozwiązania**: pasywne repeatery, bo ciężko ustalać klucz na daleką odległość

Protokoły i usługi bezpieczeństwa oraz ochrona danych poufnych i treści multimedialnych

1. Protokoły bezpieczeństwa

SSL/TLS (Secure Sockets Layer / Transport Layer Security)

- Umożliwiają bezpieczną transmisję danych w sieci (np. HTTPS).
- Stosują szyfrowanie, uwierzytelnianie i integralność danych.

IPsec (Internet Protocol Security)

- Zabezpiecza transmisję danych na poziomie sieci (np. w VPN).
- Używa szyfrowania (ESP) i mechanizmów uwierzytelniania (AH).

Kerberos

- Uwierzytelnianie użytkowników i usług w sieciach komputerowych.
- Oparte na protokole biletowym (tickets), które chronią przed atakami typu "man-in-the-middle".

SSH (Secure Shell)

- Bezpieczne połączenie zdalne do serwerów i urządzeń.
- Zapewnia szyfrowanie oraz ochronę przed podsłuchiwaniem sesji.

S/MIME (Secure/Multipurpose Internet Mail Extensions)

- Służy do szyfrowania i podpisywania cyfrowego wiadomości e-mail.

OpenID Connect i OAuth

- Zapewniają uwierzytelnianie i autoryzację w aplikacjach internetowych (np. logowanie do serwisów przez Google).

2. Usługi bezpieczeństwa

1. **Audyty i oceny bezpieczeństwa**
    - **Audyt systemów IT** – analiza aktualnego stanu zabezpieczeń, identyfikacja słabych punktów.
    - **Testy penetracyjne (Pentesty)** – symulacja ataków hakerskich w celu wykrycia luk w zabezpieczeniach.
    - **Ocena zgodności z regulacjami** – np. RODO, ISO 27001, NIS2.
    - **Ocena ryzyka** – identyfikacja potencjalnych zagrożeń i oszacowanie ich wpływu na organizację.
2. **Ochrona infrastruktury IT**
    - **Implementacja systemów ochrony**:
        1. Firewalle (zapory sieciowe).
        2. Systemy IDS/IPS (wykrywanie i zapobieganie włamaniom).
        3. Antywirusy i rozwiązania antymalware.
        4. Systemy DLP (ochrona przed wyciekiem danych).
    - **Zabezpieczenia sieciowe** – szyfrowanie danych, segmentacja sieci.
    - **Monitoring bezpieczeństwa (SOC)** – stały nadzór nad infrastrukturą IT w celu wykrycia i reagowania na incydenty.
3. **Ochrona danych**
    - **Backup i odzyskiwanie danych** – regularne tworzenie kopii zapasowych i strategie odzyskiwania danych po awariach lub atakach.
    - **Szyfrowanie danych** – ochrona wrażliwych informacji przed dostępem osób nieuprawnionych.
    - **Zarządzanie tożsamością i dostępem (IAM)** – kontrola, kto i do jakich zasobów ma dostęp.
4. **Zarządzanie incydentami bezpieczeństwa**
    - **Reakcja na incydenty (Incident Response)** – szybkie działania w przypadku naruszenia bezpieczeństwa.
    - **Analiza cyfrowa (Digital Forensics)** – badanie śladów po incydentach, w tym analizowanie złośliwego oprogramowania.
    - **Planowanie ciągłości działania (BCP)** – przygotowanie procedur na wypadek awarii lub cyberataku.
5. **Cyberbezpieczeństwo w chmurze**

    - Ochrona danych i aplikacji w środowiskach chmurowych (AWS, Azure, Google Cloud).
    - Zarządzanie tożsamością w chmurze.
    - Monitorowanie i audyt aktywności w chmurze.
6. **Szkolenia i edukacja**
    - **Edukacja pracowników**:
        1. Rozpoznawanie phishingu.
        2. Bezpieczne korzystanie z e-maila i Internetu.
    - **Szkolenia dla zespołów IT**:
        1. Zarządzanie cyberbezpieczeństwem.
        2. Obsługa systemów ochrony.
7. **Doradztwo i strategia bezpieczeństwa**
    - Tworzenie polityk bezpieczeństwa IT.
    - Planowanie strategii bezpieczeństwa długoterminowego.
    - Wdrażanie programów zgodności z regulacjami prawnymi.

3. Ochrona danych poufnych i treści multimedialnych

Dane poufne

Dane poufne obejmują informacje osobowe, dane finansowe czy tajemnice przedsiębiorstw. Ich ochrona wymaga zastosowania technologii i procedur:

- Szyfrowanie danych
    - Symetryczne (AES, DES): Szybkie szyfrowanie z jednym kluczem.
    - Asymetryczne (RSA, ECC): Użycie kluczy publicznego i prywatnego.
- Zarządzanie kluczami (Key Management)
- Systemy HSM (Hardware Security Module) i KMS (Key Management Service).
- Anonimizacja i pseudonimizacja
- Przekształcanie danych w celu ochrony prywatności.
- Mechanizmy zgodności z przepisami
- Przestrzeganie regulacji, takich jak RODO (GDPR), CCPA czy HIPAA.

Treści multimedialne

Ochrona treści multimedialnych (np. wideo, muzyki, obrazów) jest szczególnie ważna w kontekście praw autorskich i dostępu do płatnych materiałów.

- **DRM (Digital Rights Management)**
    - Kontrolowanie dostępu do treści chronionych prawem autorskim.
    - Przykłady: Widevine (Google), PlayReady (Microsoft), Denuvo.
- **Watermarking (znakowanie wodne)**
    - Dodawanie ukrytych znaków do treści multimedialnych w celu identyfikacji źródła lub właściciela.
- **Szyfrowanie transmisji strumieniowej**
    - HLS (HTTP Live Streaming) z szyfrowaniem AES-128 (dla wideo).
    - Zabezpieczenie przesyłanych danych przed nieautoryzowanym dostępem.
- **Systemy dostarczania treści (CDN)**
    - Mechanizmy ochrony przed atakami DDoS i ograniczania dostępu na podstawie lokalizacji geograficznej.

Zagrożenia i detekcja ataków sieciowych

Zagrożenia pasywne:

- Podsłuchiwanie
- Analiza ruchu sieciowego

Zagrożenia aktywne:

- Podszywanie
- Nieautoryzowany dostęp do zasobów
- Modyfikacja danych
- Blokada dostępu (DoS, DDoS)
- Malware (malicious software) to najbardziej ogólna nazwa wszelkiego rodzaju oprogramowania mającego na celu przestępcze lub złośliwe działanie w stosunku do komputera lub oprogramowania użytkownika
    a) Wirusy
    b) Robaki (w tym robaki „precyzyjne”)
    c) Konie trojańskie
    d) Rootkity
    e) Ukryte narzędzia zdalnej administracji – backdoory
    f) Aplikacje szpiegujące – spyware
    g) Exploity
    h) Programy szantażujące – ransomware

Podstawowe narzędzia ochrony:

1) Zapora sieciowa/Firewall

1) **Filtrowanie bezstanowe** (stateless firewall): pakiety, czyli trzecia warstwa modelu OSI
2) **Filtrowanie stanowe** (stateful firewall): połączenia, czyli czwarta warstwa modelu OSI
3) **Monitorowanie aplikacji** (application firewall): aplikacje/protokoły, czyli wyższe warstwy OSI

2) DMZ (strefa zdemilitaryzowana, strefa ograniczonego zaufania)

- Wydzielona strefa z odrębną polityką bezpieczeństwa, zwykle mniej restrykcyjną (serwery www, FTP, pocztowe, DNS, itp.)

3) IDS, IPS, IDPS (Intrusion Detection and Prevention System)

- **NIDS** (skanowanie ruchu sieciowego)
- **HIDS** (bada zachowanie uruchomionych procesów – plik już uruchomiono)

Główne podejścia:

1) **Sygnatury ataków**
    - Efektywne dla znanych ataków
    - „Pewne”, szybkie, precyzyjne
    - Można samemu tworzyć nowe sygnatury
    - Brak skalowalności – nawet dla podobnych ataków, wirusów polimorficznych, itp.
2) **Wykrywanie anomalii**
    - Wykrywanie nawet nieznanych ataków

    - Zwykle wymaga wzorca ruchu (trenowanie)
    - Potencjalnie skalowalne
    - Duża ilość fałszywych alarmów

4) NGFW (Next–Generation Firewall)

5) SIEM (Security Information and Event Management)

6) Honeypot

7) Antywirus

- bada zachowanie uruchomionych procesów – plik już uruchomiono
- Uruchamia plik w izolowanym środowisku nie zagrażając przy tym funkcjonowaniu systemu operacyjnego
- Rozpoznaje malware na podstawie sygnatur (suma kontrolna, określony ciąg bajtów w binarce itp.)

8) Sandbox (ochrona przez izolację, wirtualizacja zasobów)

Podstawy programowania

Język C – cechy języka, zastosowania, kompilacja, sposoby alokacji pamięci i wyciek pamięci, itp.

Cechy języka:

- Język imperatywny
- Brak wbudowanych mechanizmów ochrony pamięci
- Niski poziom – duża kontrola nad elementami blisko sprzętu (jak pamięć, kosztem np. punktu wyżej)

Zastosowania:

- Systemy operacyjne, sterowniki
- Systemy wbudowane (embedded),
- Kompilatory/interpretery,
- Oprogramowanie o wysokim stopniu optymalizacji ("ciężkie" obliczeniowo lub często wykonywane – przetwarzanie multimediów, etc.).

Kompilacja – tłumaczenie kodu źródłowego na kod maszynowy. Standardowo w 3 etapach:

1) **Preprocessing** – wstawianie plików nagłówkowych (.h), wykonywanie makr, etc.
2) **Kompilacja** – tłumaczenie kodu pośredniego (który jest wynikiem działania preprocessora) na kod maszynowy (.o)
3) **Linkowanie** – łączenie kodu obiektowego z bibliotekami zewnętrznymi, tworzenie pliku wykonywalnego.

Sposoby alokacji pamięci:

Sposób alokacji (Proces rezerwacji pamięci) pamięci zależy od rodzaju pamięci: C ma dwa typy: **pamięć statyczną** i **dynamiczną**:

- **Statyczna**:
    - C automatycznie przydziela pamięć dla każdej zmiennej podczas kompilacji programu
- **Dynamiczna**:
    - Jest przydzielana po uruchomieniu programu,
    - Pamięć dynamiczna nie należy do zmiennej, można uzyskać do niej dostęp tylko za pomocą wskaźników,
    - Funkcje takie jak malloc() i calloc() przydzielają pewną ilość pamięci i zwracają wskaźnik do jej adresu. (potrzeba <stdlib.h>),
    - Free() – dealokacja

Wyciek pamięci:

Wyciek pamięci następuje, gdy program dynamicznie alokuje pamięć i nie zwalnia jej, gdy nie jest już wykorzystywana. Może też wystąpić, gdy tracimy referencje do zaalokowanej w ten sposób pamięci.

Przyczyny:

- Nadpisanie wskaźnika bez uprzedniego zwolnienia pamięci;
- Przepełnienie bufora: Wpisywanie danych poza granice przydzielonej pamięci;
- Funkcje, które nie sprawdzają długości wprowadzanych danych. (np. sprintf())

Jak uniknąć?

- Zawsze zwalniać pamięć po jej użyciu (free());
- Zerować wskaźniki po zwolnieniu pamięci (ptr = NULL);
- Sprawdzaj długość danych przed operacjami kopiowania lub odczytu.
- Nie używać funkcji uważanych za niebezpieczne (gets, sprintf...)

Struktury danych, wskaźniki, funkcje i instrukcje

Typy danych

- typ wartości logicznych: bool, tylko w wersji C99 (po włączeniu nagłówka <stdbool.h>),
- typy całkowitoliczbowe:
    - char / unsigned char / signed char,
    - int / unsigned int,
    - short / unsigned short,
    - long / unsigned long,
    - long long / unsigned long long (C99),
- typy zmiennopozycyjne:
    - float, double, long double

Instrukcje

- Instrukcja warunkowa (if)
- Instrukcja rozgałęzienia (switch)
- Pętle
    - while
    - do while
    - for

Funkcje

<type> name(<type> arg1, <type> arg2, ...) { // deklaracja funkcji

definitions and instructions; // definicja funkcji }

Funkcja o typie różnym od void musi mieć przynajmniej jedną instrukcję **return wyr;**, gdzie wyr jest wyrażeniem zgodnym z typem funkcji. Funkcja typu void może zawierać instrukcję return; (bez wyrażenia). Jeżeli tej instrukcji nie ma to funkcja kończy się po wykonaniu wszystkich jej instrukcji.

Parametry do funkcji są przekazywane przez wartość. Oznacza to, że aktualne wartości parametrów są kopiowane do obszaru roboczego funkcji, a sama funkcja nie ma dostępu do oryginału.

Wniosek: Funkcja nie może zmienić oryginalnych wartości parametrów. Inne sposoby przekazywania parametrów są opisane w rozdziale o wskaźnikach.

Wskaźniki

Wskażnik zapisuje w pamięci adres zmiennej, do której się odnosi

int x = 10;

int *p; // Deklaracja wskaźnika do zmiennej typu int

p = &x; // Przypisanie adresu zmiennej x do wskaźnika p

printf("%d\n", *p); // Wyświetla wartość zmiennej x (10)

*p = 20; // Zmienia wartość zmiennej x na 20

Podwójne (potrójne ...) wskaźniki – wskaźnik do wskaźnika

int **ipp; // gdzie '**' oznacza dwa "poziomy" wskaźników.

Proste przykłady

int i = 5, j = 6; k = 7;

int *ip1 = &i, *ip2 = &j;

ipp = &ip1; // powoduje, że ipp wskazuje na ip1, który wskazuje na i. (*ipp jest równy ip1 a **ipp jest równy i, czyli 5).

Wskaźniki do funkcji

int (*foo)(int); \\ foo jest wskaźnikiem do funkcji posiadającej jeden argument typu int i zwracającej wartość typu int.

Deklarując wskaźniki funkcyjne postępujemy jak przy deklaracji funkcji, zamieniając nazwę funkcji, np. foo, na (*foo) (pamiętając o nawiasach).

Aby zainicjalizować wskaźnik funkcyjny musimy podać adres funkcji o parametrach zgodnych z deklaracją

int main(void) {

void (*foo)(int);

foo = &my_int_func; // the ampersand is optional

```
foo(2); // call my_int_func; you do not need to write (*foo)(2)

(*foo)(2); // but if you want to, you may

return 0;}
```

z powyższego przykładu wynika, że składnia wskaźników funkcyjnych jest elastyczna: można stosować konwencję 'wskaźnikową', z '*' i '&', albo ominąć tę część.

Struktury danych

- **Tablice** – Tablice przechowują elementy tego samego typu.

```
int arr[5] = {1, 2, 3, 4, 5};
```

- **Struktury (struct)** – Struktury umożliwiają grupowanie różnych typów danych.

```
struct Point {

int x;

int y;

};

struct Point p1 = {10, 20};
```

- **Unie (union)** – Unie przechowują różne typy danych w tym samym miejscu pamięci.

```
 union Data {

int i;

float f;

};

union Data d;

d.i = 10; // Przypisanie wartości całkowitej
```

- **Typ wyliczeniowy (enum)** – Typ wyliczeniowy definiuje zbiór nazwanych stałych.

```
enum Color { RED, GREEN, BLUE };

enum Color c = GREEN;
```

Pojęcie rekurencji i praktyczne przykłady użycia

Rekurencja to technika programowania, w której funkcja (lub procedura) wywołuje samą siebie, aż do osiągnięcia określonego warunku zakończenia, zwanego warunkiem stopu.

Rekurencja jest często stosowana do rozwiązywania problemów, które mogą być podzielone na mniejsze, podobne podproblemy.

Przykład rekurencji:

- Quicksort,
- wyszukiwanie binarne,
- dfs,
- Fibonacci,
- wieża Hanoi.

Projektowanie i analiza algorytmów

Algorytmy i ich konstrukcja oraz złożoność obliczeniowa

Popularne metody konstrukcji algorytmów:

1) **Divide and conquer** (dziel i rządź) – oryginalny problem jest dzielony na podproblemy, które są rozwiązywane rekurencyjnie.
    a. Standardowy przykład to quick sort (patrz: 2 pytania niżej).
2) **Programowanie dynamiczne** – rozbija problem na pokrywające się podproblemy, do których zapamiętywane są rozwiązania (dzięki czemu unikamy liczenia tego samego wiele razy).
    a. Przykładowy problem – znajdowanie najdłuższego wspólnego podciągu (konstruujemy rozwiązanie zapamiętując poprzednie wyniki, patrz tabela/link z niej).
3) **Algorytmy zachłanne** – w każdym kroku wybieramy optymalne rozwiązanie W DANYM MOMENCIE (lokalne optimum). Z każdym krokiem zbliżamy się do oryginalnego rozwiązania.
    a. **Uwaga** – nie we wszystkich przypadkach to da optymalne rozwiązanie (problem musi mieć postać optymalnej podstruktury i własność zachłannego wyboru).
    b. Przykładowy algorytm: Dijkstra (najkrótsze ścieżki w grafie ważonym).
4) **Rekurencyjne** – funkcja wywołuje siebie samą. Funkcja rekurencyjna musi posiadać warunek stopu, kiedy przestaje się wywoływać.
    a. Najprostszy przykład: tragiczna, rekurencyjna implementacja silni, ciągu Fibonacciego czy czegokolwiek takiego.

Dla złożoności obliczeniowej istnieje kilka notacji:

1) Big–O (O) – górne ograniczenie dla złożoności czasowej algorytmu ("najgorszy przypadek").
    a) Formalnie f(n)=O(g(n)), gdy istnieje dodatnia liczba c taka, że f(n)<=c*g(n) dla dostatecznie dużego n.
    b) Przykładowo bubble sort ma O(n^2).
2) Small–O (o) – tak jak wyżej, tylko zamiast <= musi być ściśle <
3) Big–Omega – dolne ograniczenie (najlepszy przypadek) – f(n)>=c*g(n) dla dostatecznie dużego n.
    a) Przykładowo bucket sort ma Ω(n)=n
4) Small–Omega – dolne ograniczenie ścisłe (f(n)>c*g(n), j.w.).
5) Theta (Θ) – wąskie ograniczenie, tzn. istnieją c1 i c2 takie, że c1*g(n)<=f(n)<=c2*g(n) dla dostatecznie dużego n.
    a) Przykład: merge sort Θ(n)=n logn

Analiza złożonych struktur danych

1. **Drzewa**
    - **Drzewa binarne**: Każdy węzeł ma maksymalnie dwóch potomków. Stosowane w wyszukiwaniu i sortowaniu.
    - **Drzewo BST (Binary Search Tree)**: Umożliwia szybkie operacje wyszukiwania, wstawiania i usuwania (O(log n) w przypadku zrównoważonego drzewa).
    - **Drzewo AVL**: Samobalansujące drzewo BST, gwarantujące O(log n) dla podstawowych operacji.
    - **Drzewo czerwono-czarne**: Ulepszone BST o podobnych właściwościach co AVL, ale bardziej wydajne przy wstawianiu.
    - **Drzewa B i B+**: Zoptymalizowane do operacji na dużych zbiorach danych w systemach baz danych.
    - **Drzewa trie**: Używane do przechowywania słów, np. w słownikach i wyszukiwarkach.
2. **Grafy**
    - **Grafy nieskierowane i skierowane**: Reprezentują połączenia między węzłami, np. sieci społeczne czy drogi.
        1. Reprezentacja grafów:
            1. Lista sąsiedztwa (wydajna dla rzadkich grafów).
            2. Macierz sąsiedztwa (wydajna dla gęstych grafów).
    - **Grafy ważone**: Używane w algorytmach najkrótszej ścieżki (np. algorytm Dijkstry, Bellmana-Forda).
    - **Drzewa spinające (Spanning Trees)**: Minimalne drzewa spinające (np. algorytmy Kruskala i Prima).
3. **Kolejki priorytetowe i kopce**
    - **Kopiec binarny**: Wspiera szybkie operacje znajdowania maksimum/minimum (O(log n)).
    - **Kopiec Fibonacciego**: Bardziej złożony, ale efektywny w specyficznych zastosowaniach (np. w algorytmach grafowych).
4. **Tablice z haszowaniem (Hash Tables)**
    - **Mapy i słowniki**: Umożliwiają szybki dostęp do danych w czasie O(1) (przy dobrej funkcji haszującej).
    - **Tablice mieszające z adresowaniem otwartym lub łańcuchowaniem**: Używane do obsługi kolizji.
5. **Listy połączone**
    - **Listy jednokierunkowe i dwukierunkowe**: Struktury liniowe do efektywnego dodawania/usuwania elementów.
    - **Lista cykliczna**: Używana w systemach operacyjnych (np. algorytmy planowania procesów).

Rozwijając temat:

Podczas tworzenia algorytmu ważny jest odpowiedni wybór struktury danych, który służy do przechowywania informacji. Wybrana struktura ma bezpośredni wpływ na budowę algorytmu, sposób jego implementacji, a w niektórych przypadkach również na złożoność obliczeniową. Wśród typów danych wyróżniamy proste, które służą do przechowywania jednej zmiennej np. liczby (int, float, double), znaku (char), czy logiczne (boolean), czy zmienne wyliczeniowe (enum). Wyróżniamy również podstawowe struktury danych, które przechowują proste typy danych, m.in. tablice, struktury, unie. Na bazie struktur

podstawowych buduje się struktury złożone takie jak listy, stosy, kolejki czy drzewa.

Lista:

Lista to struktura danych, której elementy są ułożone w porządku liniowym. W przypadku listy jednokierunkowej każdy element zawiera daną (którą przechowuje) i wskaźnik na następny element listy. Ostatni element w liście jako wskaźnik na następny element listy przechowuje null. Lista, której każdy element oprócz wskaźnika na następny element w liście, zawiera również wskaźnik na wcześniejszy element w liście nazywamy listą dwukierunkową. Dzięki temu można przemieszczać się w programie łatwo w obu kierunkach listy. W liście jednokierunkowej i dwukierunkowej wyróżniamy początek i koniec listy. Listę, której ostatni element wskazuje na początek listy nazywamy listą cykliczną.

Koszt dodania (na koniec listy), usunięcia (z końca listy) i znalezienia elementu w liście to O(n).

Do zalet listy należy łatwość implementacji i iterowania się przez kolejne elementy listy. Do wad należy niska elastyczność i koszt dodania i usunięcia elementu.

Stos:

Stos to liniowa struktura danych, w której dane są dokładane na wierzch stosu i z wierzchołka stosu są pobierane. Jest to bufor typu LIFO (Last In First Out). Stos można zaimplementować m.in. używając listy lub tablicy. Do podstawowych operacji wykonywanych na stosie należy:

push() - dodanie obiektu na stos - złożoność O(1)
pop() - ściągnięcie elementu ze stosu - złożoność O(1)
isEmpty() - sprawdzenie, czy stos jest pusty - złożoność O(1)

Do zalet stosu należy łatwość implementacji i efektywność operacji. Do wad stosu należy brak możliwości losowego dostępu do danych.

Kolejka:

Kolejka to struktura danych, w której nowe dane są dodawane na koniec kolejki, a z początku kolejki są pobierane dane do dalszego przetwarzania. Jest to bufor typu FIFO (First In First Out). Operacje wykonywane na kolejce są analogiczne do tych wykonywanych na stosie (push, pop, isEmpty) i mają ten sam koszt.

Do zalet kolejki należy łatwość implementacji i efektywność operacji. Do wad kolejki należy brak możliwości losowego dostępu do danych.

Drzewo:

Jest to hierarchiczna struktura danych, reprezentująca drzewo matematyczne. Drzewa ułatwiają i przyspieszają wyszukiwanie, a także pozwalają w łatwy sposób operować na posortowanych danych.

Drzewo składa się z wierzchołków (węzłów) oraz łączących je krawędzi. W drzewie wyróżniamy korzeń drzewa - wierzchołek, który nie ma rodzica. Ciąg krawędzi, który łączy dwa wybrane węzły nazywamy ścieżką. Wszystkie wierzchołki połączone z danym wierzchołkiem, a leżące na następnym poziomie

są nazywane dziećmi tego węzła. Wierzchołek, który nie ma dzieci jest liściem. Wierzchołek jest rodzicem dla każdego swojego dziecka.

Wyróżniamy różne rodzaje drzew:

- **Drzewo binarne** - drzewo, w którym każdy węzeł ma co najwyżej dwoje dzieci nazywanych lewym i prawym dzieckiem
- **BST (Binary Search Tree - binarne drzewo poszukiwań)** – jest to drzewo, w którym na lewo od danego węzła znajdują się elementy o mniejszej wartości, a na prawo znajdują się elementy o większej wartości
- **Drzewo AVL** – zrównoważone binarne drzewo poszukiwań tzn. różnica długości ścieżki od każdego liścia do korzenia drzewa może się różnić maksymalnie o 1.
- **Drzewo czerwono-czarne** – rodzaj samoorganizującego się binarnego drzewa poszukiwań. Drzewa czerwono-czarne są skomplikowane w implementacji, lecz charakteryzują się niską złożonością obliczeniową elementarnych operacji, takich jak wstawianie, wyszukiwanie czy usuwanie elementów z drzewa.

Koszt operacji w drzewie zależy od jego zrównoważenia. W drzewie zrównoważonym dodanie i usunięcie węzła w drzewie wynosi O(log n). Natomiast w drzewach niezrównoważonych jest to O(n).

Jedną ze struktur opartą na drzewie jest kopiec. Jest to struktura, w której wartości potomków węzła są w stałej relacji z wartością rodzica (na przykład wartość rodzica jest nie mniejsza niż wartości jego potomka). Dzięki temu, jeśli na “szczycie” znajduje się największa wartość, możemy z wykorzystaniem stosu zaimplementować strukturę zwracającą kolejno największe dostępne wartości. Dodanie nowego lub usunięcie istniejącego elementu ze stosu ma złożoność O(log n) i wykorzystuje się do tego proces zwany Heapify (który przywraca właściwości kopca).

Sortowanie, haszowanie, wyszukiwanie, kodowanie i kompresja

Algorytmy sortujące:

- **Sortowanie bąbelkowe** (buble sort) – Algorytm porównuje sąsiednie elementy i zamienia je miejscami, przesuwając największe elementy na koniec w kolejnych przebiegach.
- **Sortowanie przez wybieranie** (selection sort) – Algorytm wielokrotnie znajduje najmniejszy (lub największy) element z nieposortowanej części i umieszcza go na właściwej pozycji.
- **Sortowanie przez wstawianie** (insertion sort) – Algorytm wstawia elementy z nieposortowanej części na odpowiednie miejsce w już posortowanej części tablicy.
- **Sortowanie przez scalanie** (merge sort) – Algorytm dzieli tablicę na mniejsze części, sortuje je rekurencyjnie, a następnie scala w jedną posortowaną całość.
- **Sortowanie szybkie** (quick sort) – Algorytm wybiera element bazowy (pivot), dzieli tablicę na części mniejsze i większe od pivotu, a następnie sortuje je rekurencyjnie.

- **Sortowanie przez kopcowanie** (heap sort) – Algorytm buduje kopiec binarny (max–heap lub min–heap) i wykorzystuje jego właściwości do sortowania elementów.
- **Sortowanie przez zliczanie** (counting sort) – Algorytm zlicza wystąpienia każdego elementu i wykorzystuje te informacje do odtworzenia posortowanej tablicy; działa efektywnie na liczbach całkowitych w ograniczonym zakresie.
- **Sortowanie kubełkowe** (bucket sort) – Algorytm dzieli elementy na kilka przedziałów (kubełków), sortuje każdy kubełek oddzielnie, a następnie łączy wyniki.
- **Sortowanie pozycyjne** (radix sort) – Algorytm sortuje liczby w oparciu o ich cyfry, zaczynając od najmniej znaczącej do najbardziej znaczącej, stosując np. sortowanie przez zliczanie jako podprocedurę.
- **Sortowanie topologiczne** –> grafy – Algorytm stosowany w grafach skierowanych acyklicznych (DAG) do uporządkowania wierzchołków w kolejności zgodnej z zależnościami między nimi.

Haszowanie:

- Funkcja przyporządkowująca dowolnie dużej liczbie krótką wartość o stałym rozmiarze
  - Dziedzina jest wielokrotnie większa niż przeciwdziedzina (h: V T, |V| >> |T|)
- np. h(k) = k mod N, Dla ciągów znaków: suma modulo 2

Metody obsługi konfliktów:

- Metoda łańcuchowa
  - Wszystkie elementy, którym w wyniku haszowania odpowiada ta sama pozycja w tablicy T, zostają umieszczone na jednej oddzielnej liście
- Strefa przepełnienia (podział na strefę podstawową i przepełnienia)
  - Do strefy przepełnienia elementy trafiają w momencie wystąpienia konfliktu podczas zapisu danej pod indeks znajdujący się w części podstawowej. Strefa ta wypełniana jest liniowo wraz z napływem nowych elementów powodujących konflikty.
- Próbkowanie liniowe
  - W przypadku stwierdzenia konfliktu zapisujemy element na pierwsze kolejne wolne miejsce
- Podwójne haszowanie

Wyszukiwanie:

- Lista – złożoność O(n) dla posortowanej i nieposortowanej listy
- Drzewo BST – złożoność O(h)
- Drzewo przedziałowe – złożoność O(log n)
- Wyszukiwanie wzorca w tekście –> naturalne podejście – Bruteforce
- Optymalizacja –> algorytmy:
  - Knutha–Morrisa–Pratta (KMP) – Znajomość struktury wzorca pomaga optymalizować liczbę przesunięć
    - Przykładowo: jeśli pierwszy symbol wzorca więcej w nim się nie powtarza, a niezgodność jest na dalszej pozycji, nie ma sensu przesuwać tylko o jedną

    - Boyera i Moore'a – jeśli trafiamy na znak, którego w ogóle nie ma we wzorcu – można przesunąć cały wzorzec na prawo od niego, a jeśli trafiamy na znak, który we wzorcu jest – możemy przesunąć do tego znaku
    - Karpa–Rabina – wyszukuje wzorce w tekście, wykorzystując funkcję haszującą do porównywania fragmentów tekstu, co przyspiesza sprawdzanie zgodności

Kodowanie:

- **Kodowanie Huffmana**
    - Kodujemy tak, żeby symbole, które są spotykane najczęściej, miały krótsze kody. W tym celu konstruujemy optymalne drzewo kodowe
    - Potrzebujemy statystyki: częstotliwości symboli w tekście
- **Kodowanie Lempela–Ziva–Welcha (LZW)**
    - Mamy słownik, zawierający kody często występujących ciągów (lub po prostu kody znaków). Spotykając kombinację, której nie ma w słowniku, dodajemy ją do słownika i umieszczamy na wyjściu odpowiednie kody
    - Słownik jest dynamicznie tworzony dla każdego ciągu wejściowego
    - Nie musimy przesyłać słownika – algorytm dekodujący sam go odtworzy

Kompresja:

- Proces zmniejszania rozmiaru danych poprzez eliminację redundancji, co może polegać na bezstratnym zapisywaniu informacji w bardziej efektywny sposób (np. ZIP) lub stratnym usuwaniu mniej istotnych danych (np. JPEG).
- Wykorzystuje kodowanie

Wykrywanie incydentów

Pojęcie i znaczenie incydentu oraz klasyfikacja incydentów i przykładowe bazy (MITRE ATT&CK i inne)

Zdarzenie (event) – to każde widoczne wydarzenie w systemie.

Incydent – niekorzystne zdarzenie w systemie lub zagrożenie wystąpienia takiego zdarzenia (naruszenie zasad bezpieczeństwa, itp.). Znaczenie incydentów jest ogromne, ponieważ ich skutki mogą obejmować: straty finansowe, reputacyjne, sprawy prawne, itp.

Naruszenie – realne zagrożenie (rodzaj szkód; kto zdobył dostęp; czy z niego skorzystał; stopień złagodzenia ryzyka).

Klasyfikacja incydentów – kryteriami tymi są:

1) liczba dotkniętych klientów;
2) czas trwania, w tym przerwa w świadczeniu usług;
3) zasięg geograficzny;
4) utrata danych (CIA);
5) krytyczność usług, których dotyczy incydent;

6) skutki incydentu (koszty, straty, itp).

Biorąc pod uwagę te oraz inne kryteria następnie wyznaczamy macierz ryzyka.

Bazy do analizy incydentów:

- **MITRE ATT&CK** (Adversarial Tactics, Techniques, and Common Knowledge) – to szczegółowa baza wiedzy opisująca taktyki, techniki i procedury stosowane przez cyberprzestępców. Jest wykorzystywany do modelowania zagrożeń, planowania testów penetracyjnych oraz szkoleń.
- **CVE** (Common Vulnerabilities and Exposures) – dostarcza informacji o znanych podatnościach w oprogramowaniu i sprzęcie. Każda podatność jest opisana za pomocą unikalnego identyfikatora (np. CVE–2024–12345) oraz posiada krótki opis.
- **NIST Cybersecurity Framework** – framework opracowany przez Narodowy Instytut Standaryzacji i Technologii (NIST) dostarcza wskazówek dotyczących zarządzania ryzykiem cyberbezpieczeństwa. Składa się z pięciu głównych funkcji: identyfikacja, ochrona, wykrywanie, reagowanie, odzyskiwanie.
- **OWASP** (Open Web Application Security Project) – dostarcza narzędzi i zasobów skoncentrowanych na bezpieczeństwie aplikacji webowych (np. OWASP Top 10)
- **CVSS** (Common Vulnerability Scoring System) – system oceny, który pozwala na ilościowe określenie stopnia ryzyka związanego z podatnościami. Oceny CVSS są często stosowane w bazach takich jak CVE, aby określić poziom istotności podatności.

Sposoby wykrywania i reagowania na incydenty, systemy IDS/IPS i SIEM, jednostki SOC oraz działanie zespołów CSIRT i CERT

Reagowanie wg. NIS (Network and Information System) (zgodne z ISO i zaleceniami CSIRT/CERT):

Przygotowanie (ISO) –> Identyfikowanie (alerty) –>Opanowanie sytuacji (analiza/zbieranie info) –> Eliminacja (nowe IP, ograniczenie dostępów…) –> Odzyskiwanie (testy/backup) –>Raportowanie

Skuteczność obrony mierzą trzy typy wskaźników:

- rozpowszechnienia,
- efektywności,
- czasu.

Reagowanie:

- OODA – Observe Orient Decide Act – szybkie działanie
- PDCA – PLAN DO CHECK ACT (ADJUST) – nakierowanie na analizę

IDS – Intrusion Detection System – to urządzenie lub oprogramowanie, które monitoruje i analizuje ruch sieciowy w celu wyszukania podejrzanych aktywności wysyłając powiadomienia o ich znalezieniu.

- **Host–based IDS (HIDS)** – działają jako aplikacja w jednym, ochranianym systemie operacyjnym, analizując zdarzenia pochodzące z logu systemowego oraz z lokalnych interfejsów sieciowych
- **Network IDS (NIDS)** – analizujące ruch sieciowy dla wszystkich systemów w segmencie sieci, do którego są podłączone
- **PIDS** – protocol–based oraz APIDS Application Protocol–based
- Analizy sygnaturowe, oparte o anomalie oraz heurystyczne

IPS – Intrusion Prevention System – jest narzędziem zabezpieczającym sieć (może to być urządzenie sprzętowe lub oprogramowanie), które stale monitoruje sieć pod kątem złośliwej aktywności i podejmuje działania zapobiegające jej wystąpieniu, w tym zgłaszanie, blokowanie lub usuwanie jej, gdy już wystąpi.

IPS są czasami częścią next–generation firewall (NGFW) lub unified threat management (UTM).

SIEM – Security Information and Event Management – kluczowe rozwiązania pod kątem pozyskiwania informacji o zdarzeniach (każdych dotyczących przetwarzania informacji) wewnątrz firmy i monitorowania wszystkich urządzeń odpowiedzialnych za zbieranie, przetwarzanie, gromadzenie i udostępnianie informacji wewnątrz i na zewnątrz firmy. SIEM pozwala na centralne zarządzanie logami generowanymi z wielu urządzeń w jednym czasie

Składa się z:

- części sprzętowo–technologicznej w której dokonuje się gromadzenia oraz oceny zdarzeń i procesów, najczęściej dzieje się to w formie utworzenia SOC, czyli Security Operations Center
- części organizacyjno–proceduralnej, która jest uzupełnieniem części sprzętowo–technologicznej. Zawiera najczęściej wszelkie procedury bezpieczeństwa oraz regularne szkolenia wszystkich pracowników.

SOC – Security Operations Center – może być samozarządzający (w obrębie organizacji, używa SIEM), współzarządzany (SOC z zew. IR wew.), w pełni zewnętrzny (SOC i IRT z zew.)

Różnica SOC i IRT:

- IR (IRT) – reagowanie na zdarzenia, prewencja, zminimalizowanie skutków
- SOC – monitorowanie i troska o bezpieczeństwo, prewencja i edukacja

CSIRT – Computer Security Incident Response Team:

- klasyfikowanie incydentów,
- działanie informacyjne dla całego społeczeństwa,
- monitorowanie zagrożeń cyb. i incydentów na poziomie krajowym,
- szacowanie ryzyka dla całego kraju i przekazywanie inf. do innych podmiotów,
- może przygotowywać wiążące rekomendacje dotyczące urządzeń inf. lub oprogramowania (operatorzy usług kluczowych muszą się do nich zastosować),
- “Organ właściwy ma prawo zgodnie z odrębnymi przepisami powołać sektorowy zespół ds. cyberbezpieczeństwa dla danego sektora lub podsektora” –> np. akademickie, sektora wojskowego itp.

CSIRT GOV PL – prowadzony przez szefa ABW, pełni rolę Zespołu CSIRT poziomu krajowego odpowiadającego za koordynację procesu reagowania na incydenty komputerowe (NBP, incydenty związane z terroryzmem)

- Ochrona administracji publicznej: Zarządzanie bezpieczeństwem systemów teleinformatycznych w instytucjach publicznych.
- Koordynacja działań z innymi CSIRTami oraz operatorami usług kluczowych.

CSIRT MON – (Ministerstwo Obrony Narodowej):

- Ochrona infrastruktury wojskowej: Zarządzanie cyberbezpieczeństwem w systemach teleinformatycznych Sił Zbrojnych RP.

CSIRT NASK – (Naukowa i Akademicka Sieć Komputerowa):

- Wymiana informacji z organizacjami międzynarodowymi zajmującymi się cyberbezpieczeństwem.

CERT – Computer Emergency Response Team – powołało FIRST (Forum of Incident Response and Security Teams), przez które krajowe CERTy się porozumiewają

- całodobowe nadzorowanie ruchu internetowego
- podejmowanie natychmiastowych akcji w razie pojawienia się zagrożenia
- edukacja, szkolenia
- współpraca z innymi zespołami IRT (Incidents Response Team) w ramach FIRST

Etapy "cyber kill chain"

Cyber Kill Chain – model opisujący strukturę ataków („łańcuch”, bo każdy etap jest ze sobą połączony – jeśli jakiś etap się nie uda to prawdopodobnie atak się nie powiedzie):

1) **Rozpoznanie** – zbieranie jak największej ilości informacji o celu
2) **Uzbrojenie** – przygotowanie narzędzi oraz złośliwego ładunku
3) **Dostarczanie** – dostarczenie malware'u do ofiary
4) **Wykorzystanie/eksploatacja** – wykorzystanie podatności
5) **Instalacja** – zainstalowanie oprogramowania w systemie
6) **Dowodzenie i kontrola** – ustanowienie zdalnego dostępu
7) **Działania na celu** – realizacja celów ataku

Zarządzanie bezpieczeństwem informacji

Normy i standardy w zakresie budowy systemu zarządzania bezpieczeństwem informacji

Norma ISO/IEC 27001 – co zrobić?

- Międzynarodowy standard, który określa wymagania dla SZBI – Systemu Zarządzania Bezpieczeństwem Informacji (ang, ISMS – Information Security Management System)

- Obejmuje wszystkie zagadnienia dot. ochrony danych przetwarzanych przez organizację i systematyzuje polityki, instrukcje procedury, zasady i działania w tym obszarze

ISO/IEC 27002 (Kodeks dobrych praktyk zarządzania bezpieczeństwem informacji) – jak zrobić?

- Wytyczne dotyczące wdrażania środków bezpieczeństwa;
- W przeciwieństwie do ISO 27001, **NIE** podlega certyfikacji;
- Stanowi merytoryczne i opisowe rozwinięcie normy ISO/IEC 27001;
    - Zawiera szczegółowy opis i wskazówki do wdrożenia zabezpieczeń wymienionych w ISO 27001.

ISO/IEC 27003 to kolejna norma pomocnicza, która skupia się na wsparciu organizacji podczas planowania i wdrażania ISMS zgodnego z ISO 27001.

- Dostarcza wskazówek krok po kroku, jak przygotować i uruchomić system zarządzania bezpieczeństwem informacji.

Szacowanie ryzyka

Ryzyko można zdefiniować jako możliwość wystąpienia wydarzenia, które wpłynie na osiągnięcie celów, zarówno w negatywny, jak i pozytywny sposób.

Rodzaje ryzyka: rynkowe, kredytowe, projektowe, operacyjne…

Liczymy jako **prawdopodobieństwo x wpływ**: są tabelki (very high –> very low) i z tabelki się odczytuje

Ryzyko można: zaakceptować, uniknąć, przenieść, złagodzić

Szacowanie:

- kto może zostać skrzywdzony i w jaki sposób?
- co już jest robione, aby kontrolować ryzyko?
- jakie dalsze działania podjąć, aby kontrolować ryzyko?
- kto musi wykonać działanie?
- kiedy działanie jest potrzebne?

Można je złagodzić (mitigate) – wykupić ubezpieczenie, wybrać lepszą lokalizację (np. nie parter – bo powódź itp.), backupy, failover (jak padnie jeden serwer to np. drugi ciągnie, aż tamten wstanie)

Scenariusz oceny ryzyka:

- Określenie zasobów
- Określenie zagrożeń
- Identyfikacja mechanizmów bezpieczeństwa
- Wykrycie podatności
- Określenie scenariusza
- Szacowanie wpływu
- Szacowanie prawdopodobieństwa
- Szacowanie ryzyka (very high –> very low)

Identyfikacja i inwentaryzacja zasobów informacyjnych oraz audyt wewnętrzny w organizacji

Identyfikacja i inwentaryzacja zasobów

- Ma na celu określenie, jakie dane i zasoby IT (technologiczne, ludzkie, procesowe) są kluczowe dla organizacji oraz ich lokalizacji, właścicieli, wartości i poziomu wrażliwości.
- Określenie zasobów (dane, systemy ludzie)
- Kategoryzacja zasobów (jawne/poufne, operacyjne/strategiczne itp.)
- Dokumentacja zasobów
- Analiza ryzyka

Audyt wewnętrzny

Wykonywany w planowanych odstępach czasu w celu zapewnienia informacji np. czy system zarządzania bezpieczeństwem informacji jest zgodny z:

- Wewnętrznymi wymogami organizacji
- Wymogami jakiejś normy
- Został skutecznie wdrożony i jest efektywnie utrzymywany

Planowanie audytu –> **Przeprowadzenie** –> **Raportowanie**

Inwentaryzacja jest podstawą audytu: audyt wymaga dokładnej wiedzy o zasobach i ich stanie.

Probabilistyka i statystyka

Zmienna losowa, wartość oczekiwana i kwantyle

Zmienna losowa:

Zmienna losowa to zmienna, której wartości są wynikiem losowego eksperymentu. Jest funkcją, która przypisuje każdemu zdarzeniu elementarnemu liczbę rzeczywistą. Zmienną losową możemy podzielić na:

- **Zmienną losową dyskretną** – przyjmuje skończoną lub przeliczalną liczbę wartości (np. liczba wyrzuconych orłów w serii rzutów monetą).
- **Zmienną losową ciągłą** – przyjmuje wartości z przedziału liczb rzeczywistych (np. czas oczekiwania na autobus).

Niech (Ω, F, P) będzie przestrzenią probabilistyczna. Zmienną losową X nazywamy funkcję określoną na przestrzeni zdarzeń elementarnych i przyjmującą wartości rzeczywiste $X: \Omega \to R$ spełniającą warunek $\{\omega : X(\omega) < x\} \in F$, czyli $\{\omega : X(\omega) < x\}$ jest zdarzeniem dla każdego $x \in R$.

Czyli np.:

- Suma wyrzuconych szóstek oczek w n rzutach kostką
- Funkcja opisująca przedział wieku wylosowanego z pewnej populacji osobnika,

Przykład 1: Rzucając 3 razy monetą możemy otrzymać 8 różnych wyników (zdarzeń elementarnych)

- (OOO), (OOR), (ORR), (ORO), (ROO), (ROR), (RRR), (RRO)

Jeżeli założymy się, że orzeł wypadnie co najmniej 2 razy, to nasz zysk X po jednym rzucie będzie wynosił:

$$X=\begin{pmatrix} 1, & gdy\ \omega\ \epsilon\ (OOO)\ (OOR)\ (ORO)\ (ROO) \\ -1, & gdy\ \omega\ \epsilon\ (RRR)\ (ROR)\ (RRO)\ (ORR) \end{pmatrix}$$

Przykład 2: Podróżny idzie na przystanek. Nie zna godziny. Wie, że jest 1 autobus na 60 minut. Można przyjąć, że Ω = [0;60]. Wtedy, jeśli X oznacza czas oczekiwania (w minutach), to X(ω) =ω.

Inaczej: Zmienna losowa to zmienna zależna od przypadków. Może być ciągła lub dyskretna (punktowa). Może być charakteryzowana za pomocą **wartości przeciętnej**, **wariancji** i **odchylenia standardowego**.

Wartością oczekiwaną (in. nadzieją, wartością średnią) zmiennej losowej dyskretnej X, przyjmującej wartości x1, x2, ... odpowiednio z prawdopodobieństwami p1, p2, ... nazywamy wartość

$$E(X)=\sum_{i=1}^{n} x_i \cdot P(X=x_i)$$, gdzie

- x_i – możliwe wartości zmiennej losowej,
- $P(X=x_i)$ – prawdopodobieństwo przyjęcia wartości x_i,

jeżeli powyższa suma jest bezwzględnie zbieżna (tzn. zbieżny jest szereg $\sum_{i=1}^{n} x_i \cdot P(X=x_i)$). Jeśli zmienna losowa dyskretna X przybiera tylko skończoną liczbę wartości x_i, to wartość EX istnieje.

Przykład: zmienna losowa + wartość oczekiwana + wariancja:

Rzucamy dwa razy symetryczną kostką do gry. Niech X oznacza zmienną losową wyrażającą sumę uzyskanych oczek. Wyznacz jej wartość oczekiwaną oraz wariancję.

Zmienna losowa może przyjmować wartości ze zbioru: $\{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$

$$|\Omega| = 36$$

Rozkład zmiennej to $(2,\frac{1}{36}), (3,\frac{1}{18}),\ldots, (12,\frac{1}{36})$ – czyli suma oczek z dwóch rzutów, prawdopodobieństwo tej sumy)

Wartość oczekiwana to suma:

$$E(X)= 2*\frac{1}{36} + 3*\frac{1}{18} +\ldots+12*\frac{1}{36}= 7$$

Natomiast wariancja to:

$$D^2(X) = E(X^2) - (E(X))^2 = 2^2 * \frac{1}{36} + 3^2 * \frac{1}{18} + \ldots + 12^2 * \frac{1}{36} - 7^2 = \frac{35}{6}$$

Kwantyle (to miara położenia) to wartości, które dzielą rozkład zmiennej losowej na określone, równe części.

Kwantyl (Q(p)), gdzie 0 < p < 1 –> $Q(p) = p * 100\%$, gdzie p to rząd kwantyla. Kwantyl odpowiada na pytanie, jaki procent danych jest mniejszy od danej wartości. Np. kwantyl rzędu ½ to mediana. Dzielimy zbiór na 2. Kwantyl ¼ (zwany kwartyl):

Najczęściej używane kwantyle:

- **Mediana (kwantyl 0.5)**: dzieli rozkład na dwie równe części.
- **Kwartyle**:
  - Q(0.25) – pierwszy kwartyl (dolny).
  - Q(0.75) – trzeci kwartyl (górny).
- **Decyle**: dzielą rozkład na 10 równych części.
- **Percentyle**: dzielą rozkład na 100 równych części (np. Q(0.95) to 95. percentyl).

Przykład: Jeśli mamy dane: $\{1, 2, 3, 4, 5\}$:

- Mediana: 3,
- Pierwszy kwartyl (Q(0.25)): 2,
- Trzeci kwartyl (Q(0.75)): 4.

Rozkłady ciągłe i dyskretne (jednostajny, Poissona, wykładniczy, normalny)

Rozkłady dyskretne

Zmienna losowa może przyjmować skończoną lub przeliczalną liczbę wartości.

1. Rozkład jednostajny dyskretny

- Każda wartość w skończonym zbiorze ma takie samo prawdopodobieństwo.

$$P(X=x) = \frac{1}{n}, \ dla \ \ x \in \{x_1, x_2, \ldots, x_n\}$$

- **Przykład:** Rzut symetryczną kostką do gry – każda z n=6 wartości (1, 2, 3, 4, 5, 6) ma prawdopodobieństwo 1/6.

2. Rozkład Poissona

- Modeluje liczbę zdarzeń zachodzących w stałym przedziale czasu lub przestrzeni, przy założeniu, że zdarzenia są niezależne.

$$P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}, \quad k = 0, 1, 2, \ldots$$

- λ: średnia liczba zdarzeń w danym przedziale.
- **Przykład:** Liczba telefonów przychodzących na infolinię w ciągu godziny.

Rozkłady ciągłe

Zmienna losowa może przyjmować dowolne wartości w określonym przedziale liczb rzeczywistych.

1. Rozkład jednostajny ciągły

- Wszystkie wartości w danym przedziale mają takie samo prawdopodobieństwo.

$$f(x)=\begin{pmatrix} \frac{1}{b-a}, & dla\ \ x\in[a,b] \\ 0, & dla\ \ x\notin[a,b] \end{pmatrix}$$

- **Przykład:** Losowanie liczby z przedziału [0, 10], gdzie każda liczba ma równą szansę.

2. Rozkład wykładniczy

- Modeluje czas oczekiwania do wystąpienia pierwszego zdarzenia w procesie Poissona.

$$f(x;\ \lambda)=\lambda e^{-\lambda x},\ \ x\geq 0$$

- λ: parametr częstości zdarzeń.
- **Przykład:** Czas oczekiwania na przyjazd autobusu na przystanek.

3. Rozkład normalny (Gaussa)

- Jeden z najczęściej występujących rozkładów w naturze i nauce.

$$f(x;\mu,\sigma^2)=\frac{1}{\sqrt{2\pi\sigma^2}}\ \frac{e^{¿}}{2\sigma^2}$$

Dla wyjaśnienia:

- μ – średnia (wartość oczekiwana).
- σ^2 wariancja (kwadrat odchylenia standardowego).
- x – wartość zmiennej losowej.

Właściwość	Dyskretny (Poissona)	Dyskretny (Jednostajny)	Ciągły (Jednostajny)	Ciągły (Normalny)	Ciągły (Wykładniczy)
Zakres	¿ ¿ ¿	$\{x1,x2,...,xn\}$	$[a,b]$	$(-\infty,\infty)$	$[0,\infty)$
Przykład	Liczba zdarzeń na godzinę	Wynik rzutu kostką	Losowa wartość w przedziale	Rozkład wzrostu w populacji	Czas między zdarzeniami

Średnia, wariancja i przedziały ufności

Średnia (średnia arytmetyczna) to podstawowa miara tendencji centralnej, która reprezentuje przeciętną wartość w zbiorze danych. Oblicza się ją, sumując wszystkie obserwacje w zbiorze i dzieląc przez ich liczbę, np.

$$\frac{X_1 + X_2 + X_3}{3}$$

Wariancja to miara rozproszenia danych wokół średniej. Pokazuje, jak bardzo poszczególne wartości różnią się od średniej. Wariancja populacyjna jest definiowana jako:

$$D^2(X) = E(X - E(X))^2 = E(X^2) - (E(X))^2$$

Wariancja istnieje, gdy całka (szereg) jest bezwzględnie zbieżna.

Przedział ufności to zakres wartości, który z określonym prawdopodobieństwem (poziomem ufności) obejmuje prawdziwą wartość parametru populacji, np. średniej. Jest to narzędzie pozwalające oszacować niepewność wynikającą z losowości próby.

Fizyka

Fala elektromagnetyczna jako nośnik informacji

Czym jest fala elektromagnetyczna?

- **Definicja:** Fala elektromagnetyczna to zmienne w czasie i przestrzeni pole elektryczne (E) i magnetyczne (B), które są ze sobą sprzężone i wzajemnie się indukują. Fala ta rozchodzi się w przestrzeni z prędkością światła ($c \approx 3 \cdot 10^8 ¿ \frac{m}{s} ¿$) w próżni.
- **Źródło:** Powstaje w wyniku przyspieszonego ruchu ładunków elektrycznych.

Fala elektromagnetyczna jako nośnik informacji

Fala elektromagnetyczna przenosi informacje dzięki modulacji, czyli celowej zmianie jej parametrów w czasie, które kodują dane. Wyróżnia się trzy główne metody modulacji:

1. **Modulacja amplitudy (AM, Amplitude Modulation):**
    - Informacja jest zakodowana w zmianach amplitudy fali nośnej.
    - Wykorzystywana w klasycznym radiu AM i niektórych systemach transmisji danych.

$$s(t) = [A + m(t)] \cdot cos(2\pi f_c t)$$

Gdzie:

- A: amplituda fali nośnej,
- $m(t)$: sygnał informacyjny,

- f_c: częstotliwość nośna.

2. **Modulacja częstotliwości (FM, Frequency Modulation):**
   - Informacja jest zakodowana w zmianach częstotliwości fali nośnej.
   - Wykorzystywana w radiu FM, systemach audio i telewizji.

$$s(t)=A\cdot cos[2\pi f_c t+\beta\cdot m(t)]$$

Gdzie:

- β: indeks modulacji.

3. **Modulacja fazy (PM, Phase Modulation):**
   - Informacja jest zakodowana w zmianach fazy fali nośnej.
   - Wykorzystywana w nowoczesnych systemach cyfrowych, takich jak Wi–Fi czy GSM.

$$s(t)=A\cdot cos[2\pi f_c t+m(t)]$$

Przykłady zastosowań fali elektromagnetycznej jako nośnika informacji

1. **Komunikacja radiowa i telewizyjna:**
   - Stacje radiowe nadają sygnały zakodowane w modulacji AM lub FM.
   - Telewizja wykorzystuje fale elektromagnetyczne do transmisji obrazu i dźwięku.
2. **Telefonia komórkowa:**
   - Informacje (głos, dane) są kodowane w falach radiowych w pasmach częstotliwości przypisanych operatorom.
3. **Sieci bezprzewodowe (Wi–Fi, Bluetooth):**
   - Dane są przesyłane za pomocą fal radiowych w zakresie mikrofal (2,4 GHz, 5 GHz).
4. **Transmisja satelitarna:**
   - Fale elektromagnetyczne umożliwiają komunikację między satelitami a Ziemią, np. w systemach GPS czy transmisji telewizyjnej.
5. **Światłowody:**
   - Fale elektromagnetyczne w zakresie światła widzialnego i podczerwieni przenoszą dane cyfrowe w formie impulsów świetlnych.

Zalety fal elektromagnetycznych jako nośnika informacji

- **Brak medium:** Mogą propagować się w próżni, co eliminuje potrzebę fizycznego przewodnika.
- **Szeroki zakres częstotliwości:** Od fal radiowych (10^3 Hz) po promieniowanie gamma (10^{20} Hz).

- **Szybkość propagacji:** Prędkość światła pozwala na bardzo szybkie przesyłanie informacji.
- **Wszechstronność:** Możliwość zastosowania w różnych technologiach (radio, telewizja, Internet, GPS).

Stan kwantowy jako sposób na zapis, transmisję i przetwarzanie informacji kwantowej

Ogólne pojęcia:

Cząstka i foton – podstawowe obiekty, za pomocą których najprościej jest zaobserwować, inicjować i utrzymywać zjawiska kwantowe.

- **Cząstka** jest to fizyczny obiekt, w obliczeniach kwantowych (wykorzystywane do tworzenia qubitów) to są, np. elektrony, fermiony czy trapped–ions (uwięzione jony? Nie wiem, jak to jest po polsku).
- **Foton** to kwant fali elektromagnetycznej ("cząsteczka światła"), również może być użyty jako qubit.

Funkcja falowa (ψ) – Opisuje stan kwantowy systemu, gdzie jej kwadrat modułu określa prawdopodobieństwo znalezienia cząstki w danym miejscu.

Równanie Schrödingera – Fundamentalne równanie opisujące, jak funkcja falowa zmienia się w czasie.

Superpozycja – własność w mechanice kwantowej, w której system kwantowy (np. cząstka, foton, ale też, np. grupa atomów) istnieje w wielu stanach równocześnie (np. cząstka jest w wielu położeniach równocześnie albo qubit przyjmuje równocześnie wartość 0 oraz 1).

- Przykład: qubit w stanie $|0\rangle+|1\rangle$

Splątanie (entanglement) – zjawisko, w którym stany kwantowe dwóch lub więcej systemów kwantowych (np. cząsteczek) są wzajemnie powiązane, niezależnie od odległości między nimi. Zmiana stanu jednej cząstki natychmiast wpływa na stan drugiej.

Teleportacja kwantowa – sposób przesyłu informacji za pomocą zjawiska splątania kwantowego poprzez fakt, że zmiana stanu jednej cząstki natychmiast wpływa na stan drugiej niezależnie od odległości. Natomiast aby zakończyć teleportację kwantową klasyczna informacja musi zostać wysłana od nadawcy do odbiorcy. A ponieważ klasyczne informacje muszą zostać wysłane, teleportacja kwantowa nie może nastąpić szybciej niż prędkość światła (dla ciekawych, to jest odpowiedź na paradoks EPR).

Zapis:

Kodowanie kwantowe (quantum encoding) – sposób zapisania informacji klasycznej za pomocą stanów kwantowych. Są różne rodzaje kodowania kwantowego, np: Basis Encoding (bit jest kodowany jako qubit) czy Amplitude Encoding (kodowanie z wykorzystaniem funkcji falowej poprzez przypisanie amplitud stanom kwantowym)

Zasada braku możliwości kopiowania stanów kwantowych (no–cloning theorem) – Fundamentalna zasada mówiąca, że nie można skopiować dowolnego, nieznanego stanu kwantowego bez jego zakłócenia.

Transmisja:

Kanały kwantowe – fizyczne medium umożliwiające przesyłanie informacji kwantowej (qubitów).

Transmisja za pomocą splątania – wykorzystanie zjawiska splątania do zabezpieczenia transmisji kwantowej (nie można odczytać ani skopiować stanu kwantowego, tym samym odpada MITM)

Przetwarzanie informacji kwantowej:

Obliczenia kwantowe – wykorzystując zjawisko superpozycji i splątania można przeprowadzić obliczenia na wielu możliwych stanach qubitów jednocześnie. To nie jest zrównoleglenie, tutaj stan kwantowy $|q_1q_2>$ może mieć jednocześnie wartość |00>, |01>, |10> i |11>

- np. operacja mnożenia stanu |10> * $|q_1q_2>$ to operacja jednoczesnego mnożenia wszystkich możliwych stanów stanu kwantowego | $q_1q_2>$ przez 2.

Pomiar kwantowy – pomiar kwantowy zaburza zjawiska kwantowe. Dlatego przetwarzając informację kwantową z wykorzystaniem zjawisk kwantowych trzeba uważać na to, żeby w jakikolwiek sposób nie naruszyć (zmierzyć) stanu kwantowego

Podstawy mechaniki kwantowej pozwalające zrozumieć istotę działania komputera kwantowego i kryptografii kwantowej

Pojęcia i elementy związane z komputerem kwantowym:

Prawdopodobieństwo kwantowe – kwadrat amplitudy funkcji falowej. W informatyce kwantowej amplitudy funkcji falowej (z których wynika prawdopodobieństwo) znajdują się przy stanach qubitów. Czyli, np. gdy qubit ma stan: $|\psi\rangle=\alpha|0\rangle+\beta|1\rangle$, to prawdopodobieństwo stanu $|0\rangle$ wynosi α_2 a prawdopodobieństwo stanu $|1\rangle$ wynosi β_2.

Normowanie funkcji falowej – funkcja falowa musi być tak skonstruowana, aby całkowite prawdopodobieństwo wynosiło 1. W informatyce kwantowej to oznacza, że całkowite prawdopodobieństwo układów kwantowych (wszystkich qubitów) musi wynosić 1. Czyli w nawiązaniu do poprzedniego przykładu, $\alpha_2 + \beta_2 = 1$

Pomiar kwantowy – pomiar kwantowy zaburza zjawiska kwantowe. Dlatego przetwarzając informację kwantową z wykorzystaniem zjawisk kwantowych trzeba uważać na to, żeby w jakikolwiek sposób nie naruszyć (zmierzyć) stanu kwantowego.

Dekoherencja kwantowa (spowodowana szumem (ang. noise)) – zniszczenie zjawisk i stanów kwantowych na skutek interakcji systemu kwantowego z otoczeniem (można to sobie wyobrazić trochę jako przybliżenie

konceptu pomiaru do stanu kwantowego, co skutkuje zaburzeniem całkowitego prawdopodobieństwa układu)

Bramki kwantowe – są to operacje na qubitach.

- Przykładowo, bramka Pauli–X (X gate) jest odpowiednikiem klasycznego NOT.
- A bramka Hadamarda wprowadza qubit w stan superpozycji.

Obwód kwantowy – zestaw qubitów i bramek kwantowych realizujących szereg operacji, np. realizujących **algorytm Shora**. Innymi słowy uruchomienie obwodu kwantowego jest podstawowym sposobem uruchomienia programu kwantowego.

Przykładowe algorytmy kwantowe (algorytmy z kryptografii kwantowej poniżej):

QFT (Quantum Fourier Transform) – służy do przekształcenia amplitud stanów kwantowych w superpozycję stanów z różnymi fazami

- Faza jest właściwością funkcji falowej kwantowej, która opisuje stan kwantowy.
- Podczas gdy funkcja falowa dostarcza informacji o prawdopodobieństwie znalezienia cząstki w określonym miejscu, faza odgrywa kluczową rolę w interferencji kwantowej.

Algorytm Grovera – to jest algorytm szybkiego wyszukiwania w nieuporządkowanej bazie danych. Jest znacznie szybszy niż klasyczne algorytmy. Jego złożoność wynosi $O(\sqrt{N})$, podczas gdy złożoność algorytmów klasycznych wynosi $O(N)$ dla bazy składającej się z N elementów.

Algorytm Deutsch–Jozsa – algorytm rozpoznający, czy funkcja f(x) jest stała (zwraca zawsze 0 lub 1 dla każdego wejścia) czy zrównoważona (przyjmuje tyle samo wartości 0 i 1). Dzięki superpozycji i interferencji, problem może być rozwiązany w jednym wywołaniu funkcji kwantowej, podczas gdy klasycznie wymagałoby $2^{n-1}+1$ wywołań. Głównym celem tego algorytmu jest pokazanie wyższości obliczeń kwantowych nad klasycznymi.

Kryptografia kwantowa:

Bezpieczeństwo kryptografii jest oparte na następujących zasadach mechaniki kwantowej:

- **Cząstki są z natury nieokreślone** – Na poziomie kwantowym cząstki mogą jednocześnie istnieć w więcej niż jednym miejscu lub w więcej niż jednym stanie jednocześnie. Niemożliwe jest przewidzenie ich dokładnego stanu kwantowego, co, np. uniemożliwia atakującemu przewidzenie przesyłanego stanu kwantowego.
- **Układ kwantowy nie może być zmierzony bez jego zakłócenia** – Zgodnie z prawami fizyki kwantowej, sam akt pomiaru lub obserwacji układu kwantowego zawsze będzie miał mierzalny wpływ na ten układ. To powoduje, że jeżeli atakujący zmierzy przesyłany układ kwantowy, to automatycznie się on zapada, co można wykryć. Atakujący również nie jest w stanie “podrobić”/odtworzyć przechwyconego układu kwantowego, co eliminuje MITM

- **Cząstki mogą być częściowo, ale nie całkowicie sklonowane (no–cloning theorem)** – Podczas gdy właściwości niektórych cząstek mogą być sklonowane, uzyskanie 100% identycznej kopii uważa się za niemożliwe. Przez to atakujący nie może sklonować przechwyconego stanu kwantowego, żeby go później odczytać albo żeby przeprowadzić na nim operacje

Klucze kwantowe (quantum key distribution, QKD) – proces wymiany tajnych kluczy z wykorzystaniem mechaniki kwantowej, odporny na podsłuchy.

Protokół BB84 – pierwszy kwantowy protokół kryptograficzny, jest to protokół QKD. Jego działanie polega na tym, że:

- najpierw Nadawca (Alice) wysyła fotony o polaryzacji zgodnej z losowo wybranymi bazami (np. podstawową lub ukośną).
- Odbiorca (Bob) mierzy fotony, również używając losowo wybranych baz.
- Po transmisji Alice i Bob publicznie porównują użyte bazy (bez ujawniania wyników pomiarów).
    - Zatrzymują tylko te, gdzie ich bazy były zgodne. Wyniki pomiarów w zgodnych bazach są używane do stworzenia wspólnego klucza.
- Gdy ewentualny podsłuchiwacz (Eve) przeprowadza pomiar fotonów, nie zna baz Alice ani Bob, więc wprowadza błędy do transmisji.
    - Alice i Bob wykrywają te błędy poprzez porównanie części klucza w sposób publiczny.
- Jeśli liczba błędów przekroczy próg, transmisja zostaje uznana za narażoną i klucz nie jest używany. Ponadto Eve nie może skopiować przesyłanych fotonów, żeby potem je odczytać wykorzystując opublikowane przez Alice i Boba bazy, ze względu na no–cloning theorem.

Protokół E91 – to jest również protokół QKD, ale wykorzystujący zjawisko splątania kwantowego. wikipedia

Algorytm Shora, algorytm Regeva – są to algorytmy służące do faktoryzacji naturalnych półpierwszych liczb. Zostały opracowane w celu złamania algorytmów klucza publicznego, w szczególności algorytmu RSA.

- Złamanie algorytmu RSA prowadziłoby do utraty bezpieczeństwa systemów korzystających z tej technologii, takich jak systemy płatności bankowych, mechanizmy ochrony prywatności w Internecie, systemy weryfikacji integralności plików, procedury uwierzytelniania oraz inne kluczowe rozwiązania w zakresie bezpieczeństwa danych.

Algorytm Regeva powstał w sierpniu 2023 r. i jest wielowymiarową wersją **algorytmu Shora**. Algorytm Regeva adresuje problem dużej ilości wykorzystywanych bramek kwantowych przez algorytm Shora (duża ilość wykorzystywanych bramek skutkuje większym szumem i tym samym dekoherencją kwantową, czyli zniszczeniem stanów kwantowych i uniemożliwieniem działania algorytmu). Innymi słowy ten algorytm ma przyspieszyć moment, w którym faktoryzacja dużych liczb półpierwszych z wykorzystaniem komputera kwantowego stanie się możliwa.

Ataki na kryptografię kwantową:

1) **Atak na kanał optyczny**:

    - Podsłuchiwacz wprowadza zakłócenia w przesyłane fotony (np. przez użycie własnych fotonów), próbując odczytać klucz.
    - **Detekcja** przez analizę błędów.
2) **Atak PNS (Photon Number Splitting)**:
    - Podsłuchiwacz przechwytuje wielofotonowe stany wysłane przez Alice, zachowując część fotonów dla siebie, by odczytać klucz bez wzbudzenia podejrzeń.
    - **Obrona**: Zastosowanie źródeł pojedynczych fotonów.
3) **Atak na detektory (Detector Blinding Attack)**:
    - Podsłuchiwacz oślepia detektory Boba intensywnym światłem, by wymusić określony wynik pomiaru.
    - **Obrona**: Stosowanie detektorów odpornych na oślepianie.
4) **Atak imitujący urządzenia (Trojan Horse Attack)**:
    - Wysyłanie sygnałów świetlnych do urządzenia Alice lub Boba w celu odczytania konfiguracji baz lub wyników pomiaru.
    - **Obrona**: Filtry optyczne i detekcja nieautoryzowanych sygnałów.

Systemy operacyjne

Programy, procesy i wątki w systemie operacyjnym oraz pojęcie stosu i sterty w kontekście pamięci wirtualnej systemu operacyjnego

Program komputerowy

- Sekwencja lub zestaw instrukcji w języku programowania, które komputer wykonuje.
- Składnik oprogramowania (np. wraz z dokumentacją).
- **Kod źródłowy**: forma czytelna dla człowieka.

Proces tworzenia programu:

1. Kod źródłowy -> Kompilacja -> Linkowanie -> Kod binarny.

Proces uruchamiania programu:

1. Kod binarny (program) -> Wczytanie do pamięci operacyjnej -> Nadanie PCB (Process Control Block) -> Proces.

Rola systemu operacyjnego:

- Program sterujący: nadzoruje procesy, obsługuje urządzenia I/O, przechwytuje błędy (np. w systemach wbudowanych).

Proces

- **Definicja:** Wykonywany program.
- **Cechy:**
    - Aktywny – licznik rozkazów wskazuje kolejną instrukcję.
    - Ma własny PCB, stos oraz przestrzeń adresową.
    - Jest izolowany – zablokowanie jednego procesu nie wpływa na inne.

    - Proces macierzysty nie wpływa na potomne.

Charakterystyka procesu:

1. Właściciel zasobów (przestrzeń adresowa, obraz procesu).
2. Planowanie i wykonywanie (przeplatanie z innymi procesami).

Stany procesu:

- New, Ready, Running, Waiting/Blocked, Terminated.

Zarządzanie procesami:

- Obejmuje kontrolę i separację.
- System operacyjny przełącza procesy, co zwiększa efektywność.
- Procesy są czaso- i zasobochłonne:
    - Tworzenie, terminacja, przełączanie kontekstu.

Elementy procesu w pamięci:

1. **Stos**:
    - Tymczasowe dane (zmienne lokalne, wywołania funkcji).
    - LIFO (Last In, First Out), automatyczne zarządzanie.
    - W momencie wyjścia z funkcji, pamięć stosu jest zwalniana

1. **Sterta**:
    - Pamięć dynamiczna.
    - Ręczne zarządzanie (np. malloc/free w C), możliwość wycieków pamięci.
    - Zazwyczaj większa niż stos

1. **Sekcja danych**:
    - Zmienne globalne i statyczne.
        1. Dane, których program używa podczas wykonywania

1. **Sekcja tekstu**:
    - Instrukcje programu do wykonania.

Alokacja pamięci sterty nie jest tak bezpieczna jak alokacja pamięci stosu, ponieważ dane przechowywane w tej przestrzeni są dostępne lub widoczne dla wszystkich wątków. Może dojść do wycieku pamięci.

Wątek

- **Definicja:** Podstawowa jednostka wykorzystania procesora, część procesu (Lightweight process).
- **Cechy:**
    - Szybsze i mniej zasobożerne niż procesy.
    - Bezpośrednio współdzieli zasoby procesu (sekcję kodu, dane, pliki).
    - Ma własny identyfikator (TID), licznik programu, rejestry i stos.

Charakterystyka wątku:

- Wątek jednowątkowego procesu: PID == TID.
- Tworzenie wątków nie wymaga wywołań do jądra.
- Przełączanie wątków odbywa się szybciej niż procesów.
- Zablokowanie procesu blokuje wszystkie jego wątki.

- Zmiany w procesie wpływają na wątki tego procesu.

Zastosowanie wielowątkowości:

1. Serwer WWW i przeglądarka internetowa.
2. Serwer poczty elektronicznej.
3. Funkcje wspierające, np. sprawdzanie pisowni w edytorach tekstu.

Zalety wielowątkowości:

- Responsywność.
- Współdzielenie zasobów.
- Skalowalność.
- Efektywne wykorzystanie architektury wielordzeniowej.

Pamięć wirtualna

- **Definicja:** Technika umożliwiająca wykonywanie procesów bez pełnego załadowania do pamięci fizycznej.
- **Cechy:**
  - Oddziela pamięć logiczną od fizycznej.
  - Współdzielenie plików, bibliotek, pamięci dzielonej.
  - Wymaga starannej implementacji – błędy mogą obniżać wydajność.

Podsumowanie różnic między programem a procesem

- **Program**: Pasywny – plik wykonywalny zawierający listę instrukcji.
- **Proces**: Aktywny – wykonywany program w pamięci RAM.

Techniki komunikacji międzyprocesowej

Zastosowania komunikacji międzyprocesowej:

- Współdzielenie informacji
- Przyspieszenie obliczeń
- Modularność oprogramowania

Trzy metody dzielenia informacji:

- Przez filesystem
- Przez kernel (shared info) –> message passing (kolejka wiadomości)
  - Użyteczny do wymiany niewielkich ilości danych (brak konieczności zapobiegania konfliktom)
  - Łatwiejszy w implementacji
- Przez shared memory
  - Najszybsza możliwa komunikacja
  - Jedyna wymagana interwencja z kernela to utworzenie tej pamięci

Implementacje IPC (Inter–process communication):

- Wymiana przez pliki
- Pamięć dzielona – wymaga mechanizmów synchronizacji, aby uniknąć konfliktów dostępu.
- Sygnały
- Semafory – Mechanizm synchronizacji, który służy do kontrolowania dostępu do wspólnych zasobów przez różne procesy.

    - Pozwalają na unikanie wyścigów i zapewniają bezpieczeństwo w dostępie do współdzielonych zasobów.
- Potoki nazwane lub nienazwane – Umożliwiają komunikację między dwoma procesami poprzez przesyłanie danych w jednym kierunku, z jednego procesu do drugiego. Może być anonimowa (do procesów w tym samym systemie) lub nazwana (może łączyć procesy na różnych maszynach).
- Kolejki – Umożliwia to asynchroniczną komunikację, zapewniając bufor między procesami.
- Gniazda dziedziny UNIX
- Gniazda UDP/TCP
- RPC
- niezależne lub współbieżne współdzielenie informacji, przyspieszenie obliczeń

Gniazda sieciowe, komunikacja TCP/IP pomiędzy dwoma systemami komputerowymi

Gniazdo (ang. socket) – unikalna struktura dwóch par liczb (czyli czterech liczb):

- adres IP i port klienta
- adres IP i port serwera

Sieci komputerowe:

- Warstwa łącza: Ethernet, SLIP, PPP
- Protokoły sieciowe: IPv4, IPv6, ICMP
- Protokoły transportowe: TCP (uzgodnienie trójfazowe (SYN -> SYN+ACK -> ACK)), UDP, SCTP
- Protokoły aplikacji: DNS, HTTP, SSH, etc.

Ważne funkcje:

- socket()
- bind()
- listen()
- connect()
- accept()
- write()
- read()

Programowanie skryptowe

Python jako język skryptowy (model obiektowy, paradygmat programowania funkcyjnego, popularne implementacje, itp.)

Python

- Wysokopoziomowy, dynamicznie typowany, interpretowany język programowania ogólnego przeznaczenia

- Jego uniwersalność i łatwość użycia sprawiają, że jest szeroko wykorzystywany zarówno jako język skryptowy, jak i narzędzie do budowy złożonych aplikacji.

Model obiektowy:

- Python jest w pełni obiektowym językiem programowania, w którym wszystko jest obiektem, włącznie z typami prostymi (liczby, napisy, listy itd.)
- Python wspiera tworzenie klas i obiektów, pozwalając na definiowanie własnych struktur danych
- Python obsługuje zarówno pojedyncze, jak i wielokrotne dziedziczenie, pozwalając na tworzenie złożonych hierarchii klas.

Paradygmat programowania funkcyjnego:

Programowanie funkcyjne w Pythonie realizowane jest dzięki:

- **funkcjom wyższego rzędu**,
- **anonimowym funkcjom (lambda)**,
- narzędziom takim jak **map**, **filter** czy **itertools**.

Paradygmaty:

- obiektowe
- proceduralne
- funkcyjne

Popularne implemetacje:

- **CPython** to domyślna i najbardziej powszechna implementacja.
- **PyPy** jest szybki dzięki JIT, idealny do obliczeń.
    - JIT (ang. just–in–time compilation) – metoda wykonywania programów komputerowych polegająca na ich kompilacji do kodu maszynowego bezpośrednio przed wykonaniem danego fragmentu kodu.
- **Jython** integruje Python z Javą, ale ma ograniczenia wydajnościowe i kompatybilnościowe.

JavaScript, HTML i CSS

JavaScript

Język programowania pozwalający dodawać interaktywność i dynamikę do stron internetowych. Jest to język interpretowany, wykonywany po stronie klienta przez przeglądarkę użytkownika.

- **Kluczowe zastosowania JavaScript obejmują**:
    - **Manipulacja DOM (Document Object Model**) – np. dynamiczna zmiana treści, dodawanie i usuwanie elementów.
    - **Walidacja danych w formularzach** (oczywiście można ją ominąć) – poprawa UX i zmniejszenie obciążenia serwera.
    - **Tworzenie animacji i efektów interaktywnych** – np. rozwijane menu, efekty przejścia.

    - **Obsługa zdarzeń** – reagowanie na akcje użytkownika, takie jak kliknięcia czy ruchy myszy.

W połączeniu z bibliotekami i frameworkami, takimi jak React, Angular czy Vue.js, JavaScript pozwala budować pełnoprawne aplikacje webowe typu SPA (Single Page Applications).

HTML (HyperText Markup Language)

- Język znaczników używany do tworzenia struktury i semantyki dokumentu webowego
- Plik HTML składa się z dwóch głównych sekcji:
    - nagłówka (<head>)
    - ciała dokumentu (<body>)

CSS (Cascading Style Sheets)

- Odpowiada za wygląd i stylizację stron internetowych.

Realizacja asynchronicznych zapytań HTTP i bezpieczeństwo połączeń HTTP

Asynchroniczne zapytania HTTP umożliwiają wydajną obsługę wielu operacji w tym samym czasie, co jest szczególnie przydatne w aplikacjach webowych, w których użytkownicy oczekują szybkiej reakcji.

Umożliwiają realizację zapytań HTTP bez blokowania głównego wątku aplikacji.

Np. w JS można użyć:

- fetch wykonuje zapytanie HTTP.
- async/await sprawia, że kod czeka na zakończenie operacji asynchronicznej bez blokowania aplikacji.

Bezpieczeństwo HTTP:

- **HTTPS** – TLS używa i szyfruje dane
- **HSTS** – wymusza używanie HTTPS
- **HTTPOnly** – blokuje próby odczytu cookie z tą flagą przez API inne niż HTTP
- **SameSite Strict** – w przypadku, gdy zapytanie do serwera zostanie wygenerowane z innej domeny (zapytanie typu cross–site) niż ta, dla której ciasteczko zostało utworzone, przeglądarka nie dołączy takiego ciasteczka do żądania.
- **X–Frame–Options** – zapobiega clickjackingowi – może nie wyrenderować strony

Bezpieczeństwo lokalnych sieci komputerowych

Segmentacja sieci w warstwie 2 modelu OSI/ISO

VLANy (802.1Q) – podział na logiczne grupy wewnątrz jednej sieci

- Rozgranicza ruch na poziomie warstwy 2
- Ogranicza domenę rozgłoszeniową

Najważniejsze elementy w VLANach:

- Id (od 1 do 4095), 1 to native VLAN, warto zmieniać, żeby uniknąć VLAN hoppingu itd.
- Trunki – port, który konfigurujemy między dwoma switchami lub switchem i ruterem, który przenosi ruch należący do wielu VLANów
- Access porty – porty które konfigurujemy pod urządzenia końcowe dla 1 VLANu

Bardziej szczegółowy podział (choć to chyba bardziej zakres z 4–5 semestru)

Wewnątrz VLANów możemy definiować secondary VLANy **(community** i **isolated)**:

- Urządzenia z tego samego community VLANu mogą się komunikować,
- Isolated nie mogą się komunikować z żadnym innym urządzeniem.

Do tego przy secondary VLANach mamy porty **promiscuous**, które może komunikować się ze wszystkimi urządzeniami z tego primary VLANu.

Zalety:

- Zwiększenie wydajności sieci,
- Poprawa bezpieczeństwa osiągnięta przez izolację,
- Skalowalność.

Ataki:

- **VLAN Hopping** (przeskakiwanie między VLAN–ami)
- **Spoofing** protokołów zarządzających VLAN–ami
- **Flooding CAM** (Content Addressable Memory) – Przepełnienie tablicy adresów MAC w przełączniku, co zmusza przełącznik do rozgłaszania ruchu w całej sieci
    - Atakujący wysyła dużą liczbę ramek Ethernet, każda z różnymi, losowymi adresami MAC jako źródłami.

Protokoły zapewnienia niezawodności w sieciach Ethernet

Protokoły:

- **Ethernet CSMA/CD** (Carrier Sense Multiple Access with Collision Detection): Zapewnia detekcję kolizji i retransmisję w przypadku wykrycia kolizji w trakcie transmisji.

- **Spanning Tree Protocol (STP)**: Zapewnia redundancję i unika pętli w sieci Ethernet poprzez dynamiczne wyłączanie nieaktywnych ścieżek. Jeśli jedno z aktywnych łączy ulegnie awarii no to protokół STP uruchomi alternatywne połączenie
- **Link Aggregation (LACP – Link Aggregation Control Protocol)**: Umożliwia łączenie kilku fizycznych połączeń w jeden logiczny link w celu zwiększenia przepustowości i niezawodności. Umożliwia wykrywanie awarii łącza i automatyczne przełączanie na inne dostępne łącze
    - Od Cisco - **EtherChannel**
- **Ethernet Frame Check Sequence (FCS)**: Zapewnia wykrywanie błędów transmisji na poziomie ramki poprzez stosowanie sumy kontrolnej.
- **MSTP** jest rozszerzeniem STP, które pozwala na lepsze wykorzystanie zasobów sieciowych, oferując możliwość tworzenia wielu drzew rozpinających przypisanych do różnych VLAN–ów

Przykłady ataków:

- **STP Manipulation**: Atakujący może podszyć się pod fałszywego root bridge'a, zmieniając priorytet w STP i zmieniając topologię sieci. W wyniku tego ataku mogą powstać pętle, które zakłócą komunikację w sieci.
    - Użycie funkcji "Root Guard" na przełącznikach, które zapewnia, że tylko zaufany przełącznik może pełnić rolę root bridge'a.
- **Jamming**: Atakujący może generować zakłócenia sygnału, które uniemożliwią wykrycie wolnego kanału i spowodują kolizje.

Obrona przed atakami:

- Zastosowanie szyfrowania
- Monitorowanie i wykrywanie anomalii
- Segregacja sieci i segmentacja
- Zabezpieczenie STP
    - Wdrażanie funkcji Root Guard.

Cyberbezpieczeństwo w sieciach Ethernet

1. Port Based Traffic Control

a) Storm Control – mechanizm monitorujący i kontrolujący ilość ruchu sieciowego na danym interfejsie przełącznika.

Zakres działania:

- Monitorowane są trzy typy ruchu: unicast, multicast i broadcast.
- Ruch jest przeliczany na liczbę ramek na sekundę (frames per second – FPS) lub na ilość bitów na sekundę (Traffic Rate).
- Na tej podstawie przełącznik oblicza procentową wartość przepustowości interfejsu (Bandwidth – BW), jaki jest wykorzystywany.

Możliwości konfiguracji:

- Można ustalić limity dla liczby ramek na sekundę lub BW dla każdego rodzaju ruchu (unicast, multicast, broadcast).
- Reakcje na przekroczenie limitu:
    - Blokada ruchu: Ruch zostaje zatrzymany, a interfejs przechodzi w stan err–disabled.

    - SNMP Trap: Przełącznik wysyła powiadomienie do administratora o przekroczeniu ustalonych limitów.

Falling Threshold:

- Jest to dolna granica BW lub liczby ramek na sekundę, którą ruch musi osiągnąć, aby interfejs mógł wznowić działanie po wcześniejszym przekroczeniu limitu.
- Jeśli nie zostanie skonfigurowana, wartość Falling Threshold równa się wartości górnego limitu.

b) Protected Ports – mechanizm zabezpieczający komunikację między portami na jednym przełączniku.

Działanie:

- Porty oznaczone jako "protected" nie wymieniają między sobą ruchu, niezależnie od mechanizmu flooding.
- Mechanizm działa tylko w obrębie jednego przełącznika i nie wpływa na ruch z/do innych przełączników.

Ograniczenia:

- Ruch z zewnątrz jest przekazywany normalnie, bez blokad.

c) Port Blocking – funkcja blokująca mechanizm flooding dla ramek z nieznanymi adresami unicast lub multicast.

Zastosowanie:

- Jeżeli port, na którym ustawiono Port Blocking, próbuje przesyłać takie ramki, ruch jest zatrzymywany.

d) Port Security – pozwala na nakładanie ograniczeń na ruch przychodzący przez dany port.

Możliwości konfiguracji:

- Można ustalić listę dozwolonych adresów MAC lub limit liczby adresów MAC, jakie przełącznik może się nauczyć na danym porcie.
- W przypadku naruszenia tych zasad uruchamiany jest mechanizm Security Violation.

Tryby reakcji na naruszenie:

1) **Protect**:
    - Ramki z nieznanymi adresami źródłowymi są dropowane.
    - Brak powiadomienia o naruszeniu bezpieczeństwa.
2) **Restrict**:
    - Działa jak Protect, ale dodatkowo wysyłane jest powiadomienie o naruszeniu.
3) **Shutdown**:
    - Interfejs zostaje wyłączony (err–disabled).
    - Aby przywrócić działanie, należy wykonać polecenia shutdown → no shutdown.
4) **Shutdown VLAN**:
    - Blokada dotyczy tylko danego VLAN–u, a nie całego portu.

    - Przywracanie działania wymaga restartu VLAN–u (shutdown VLAN → no shutdown VLAN).

2. Port Based Security Authentication (IEEE 802.1x)

Standard 802.1x umożliwia kontrolę dostępu do sieci i uwierzytelnianie urządzeń podłączonych do portów przełącznika.

Zakres działania:

- Obsługuje zarówno porty przewodowe, jak i bezprzewodowe.
- Uwierzytelnienie odbywa się przy użyciu serwera RADIUS.

Dostęp nieautoryzowany:

- Urządzenia nieautoryzowane mają ograniczony dostęp wyłącznie do protokołów takich jak:
    - **EAPoL (Extensible Authentication Protocol over LAN)**
        - Protokół uwierzytelniania na warstwie 2.
    - **CDP (Cisco Discovery Protocol)**.
    - **STP (Spanning Tree Protocol)**.
- Po pełnym uwierzytelnieniu urządzenie uzyskuje pełen dostęp do sieci.

b) Role w procesie uwierzytelniania:

Client (Supplicant):

- Urządzenie żądające dostępu do sieci.
- Musi obsługiwać EAPoL i mieć odpowiednie oprogramowanie.

Authenticator:

- Przełącznik brzegowy kontrolujący dostęp do sieci.
- Przekazuje ramki EAP do serwera RADIUS w formacie RADIUS.

Authentication Server:

- Serwer RADIUS, który:
    - odbiera ramki RADIUS,
    - przetwarza EAP,
    - decyduje o przyznaniu lub odmowie dostępu.

c) Wyniki procesu uwierzytelniania:

1) **Success**:
    - Pełny dostęp do sieci.
    - VLAN przypisany do portu klienta.
2) **Invalid Identity**:
    - Możliwe przypisanie klienta do ograniczonego VLAN–u.
3) **Lack of RADIUS**:
    - Po czasie oczekiwania możliwy fallback na uwierzytelnianie alternatywne (np. MAC–based).
4) **Authentication Timeout**:
    - Po upływie czasu na odpowiedź uruchamiane jest zapasowe uwierzytelnianie lub przypisanie do ograniczonego VLAN–u.

d) Host Mode w 802.1x:

Single Host: Jeden host przypisany do jednego portu.

Multiple Host: W sieciach bezprzewodowych, po uwierzytelnieniu wszystkie urządzenia trafiają do jednego VLAN–u.

3. ACL (Access Control List)

a) Charakterystyka:

ACL to narzędzie do filtrowania ruchu w oparciu o określone warunki.

- Składa się z sekwencji reguł permit lub deny.
- Domyślna reguła na końcu listy to deny any.

b) Typy ACL:

1) **Port ACL (PACL)**:
   - Stosowane na fizycznych portach przełącznika.
   - Filtrowanie tylko ruchu przychodzącego.
   - Na trunkach: obejmuje wszystkie VLAN–y.
2) **Router ACL**:
   - Stosowane na wirtualnych interfejsach (SVI), fizycznych interfejsach routerów lub LAG w warstwie trzeciej.
   - Każda ACL może być przypisana do jednego interfejsu i jednego kierunku (wejście lub wyjście).

c) VLAN Maps:

Mechanizm kontroli ruchu między urządzeniami w tym samym VLAN–ie.

Kryptografia

Kryptografia symetryczna i asymetryczna (cechy, wyzwania, stosowane szyfry blokowe, przykładowe algorytmy, itp.)

Zasada Kerkhoffsa – Bezpieczeństwo kryptosystemu nie może opierać się na nieznajomości metody szyfrowania.

Szyfr – funkcja matematyczna (algorytm) służąca do szyfrowania i deszyfrowania

Klucz – porcja wiadomości sterująca działaniem szyfru

Szyfry blokowe – szyfrują blok wejściowy o zadanej długości (jeśli tekst jawny jest krótszy, mogą to np. dopełniać zerami) w zaszyfrowany blok wyjściowy o tej samej długości;

- dają zawsze dla tych samych danych wejściowych (blok, klucz) to samo wyjście;
- zbyt mały blok to problemy z bezpieczeństwem (zachowuje strukturę tekstu jawnego),
- zbyt duży blok to problemy z wydajnością i implementacją

Szyfry strumieniowe – szyfrują jeden bajt/bit/słowo/znak tekstu jawnego na raz, korzystając ze zmieniającego się w czasie przekształcenia (nie rozważamy końca danych = uznajemy, że są nieskończone);

- szybkość, prostota, brak trybów, wektorów inicjalizacyjnych, itp.;
- np. szyfr Vigenere'a, Vernama

Kryptografia symetryczna – do szyfrowania i odszyfrowania używany jest ten sam klucz kryptograficzny:

a) **cechy**: szybkość, skalowalność, jeden klucz, służy do szyfrowania dużych danych
b) **wyzwania**: bezpieczna dystrybucja klucza
c) **szyfry blokowe**: AES, DES, 3DES
d) **algorytmy**: AES, Blowfish, RC4

Ogólna budowa szyfru AES:

- **Symetryczny** szyfr blokowy
- **Rozmiar bloku**: 128 bitów
- **Zmienna długość kluczy**: 128, 192, 256
- **Zmienna liczba rund zależna od wielkości klucza**: (Każda runda wprowadza dodatkowy poziom złożoności, utrudniając przeciwnikowi odwrócenie procesu szyfrowania bez znajomości klucza.)
    - **AES–128** – 10 rund
    - **AES–192** – 12 rund
    - **AES–256** – 14 rund

Kryptografia asymetryczna – do szyfrowania używany jest klucz publiczny odbiorcy, a do odszyfrowywania komplementarny klucz prywatny:

a) **cechy**: brak potrzeby dystrybucji klucza, podpisach cyfrowych, używany w celu uzgadnianie wspólnego sekretu (np. wymiany klucza symetrycznego)
b) **wyzwania**: wydajność, skalowalność, złożoność obliczeniowa
c) **szyfry blokowe**: RSA, ElGamal
d) **algorytmy**: RSA, Diffie–Hellman, ECC

! **Klucze dla algorytmów kryptografii symetrycznej są krótsze niż dla asymetrycznej**, ponieważ bezpieczeństwo asymetrycznych opiera się na trudnych problemach matematycznych (np. **RSA – faktoryzacja dużych liczb pierwszych**).

! **Przeważnie stosuje się połączenie obu podejść** – np. TLS, przy użyciu kryptografii asymetrycznej wymieniamy się z serwerem kluczem symetrycznym, a następnie wszystkie dane są szyfrowane przy jego użyciu.

Kryptograficzne funkcje skrótu

Funkcje skrótu (haszujące):

- Pozwalają zmapować nieskończony zbiór na skończony,
- Z wejścia o dowolnej wielkości, wyjście (hasz) zawsze będzie miało stałą długość (dla danego algorytmu),
- Są jednokierunkowe (Łatwo wylicz hasz z hasła; trudno znaleźć hasło znając tylko skrót),

- Skrót powinien silnie zależeć od każdego bitu argumentu (zmienisz jedną literę w haśle i dostaniesz zupełnie inny hasz)

Mamy różne algorytmy haszujące, m.in.: MD5, SHA–1, SHA–256, SHA–3.

UWAGA: Kolizje – siłą rzeczy mapując nieskończony zbiór na skończony, znajdą się 2 takie wartości, które dadzą ten sam hash.

- Ważne, żeby nie było za łatwo znaleźć dwie takie wartości (odporność na **atak metodą dnia urodzin** – w uproszczeniu policzenie dużej liczby skrótów i wtedy jest aż 50% szans, że trafimy na kolizję)

Dobra funkcja skrótu powinna spełniać warunki:

- **Słaba odporność na kolizje** – mając $hasz_1$ trudno znaleźć inny argument, dla którego $hasz_2$ będzie taki sam
- **Silna odporność na kolizje** – trudno znaleźć dowolną parę wiadomości, które dadzą taki sam hasz

Zastosowania funkcji skrótu:

- Kontrola integralności plików
- Podpisy cyfrowe (liczymy skrót z pliku i podpisujemy kluczem prywatnym. Odbiorca może odszyfrować załączony hasz za pomocą klucza publicznego, policzyć samemu hasz z pliku i porównać te wartości – jeśli są równe to potwierdziliśmy, że to właściciel klucza wysłał ten plik)
- Bezpieczne przechowywanie haseł w systemie.
    - Np. w systemie Linux hasła są przechowywane w formie haszu.
- Generatory liczb pseudolosowych
- Blockchain i kryptowaluty

Jak złamać hasz?

- Złamać kryptograficznie (np. MD5, dlatego nie jest uznawany za bezpieczny)
- Bruteforce – lecimy po kolei wszystkie możliwe opcje, strasznie czasochłonne, dlatego im dłuższe hasło, tym dłużej będzie łamane przez bruteforce
    - Chyba że dojdzie do kolizji haszów, albo jakiś debil w Microsoft uzna, że jak hasło jest za długie to je będziemy obcinać w ich algorytmie haszującym
- Atak słownikowy – mamy słownik z popularnymi hasłami i liczymy hasz dla wszystkich tych haseł albo dodatkowo modyfikujemy jakoś te hasła np. zmieniając „A” na „4” itp.

Apropo przechowywania haseł warto wspomnieć o kilku rzeczach:

- **Sól** – losowy ciąg bitów doklejany do hasła przed zahashowaniem, żeby te same hasła nie dawały tego samego haszu. Sól jest przechowywana obok hashu hasła.
    - Sól powinna mieć conajmniej 128 bitów, aby uniknąć używania tej samej soli dla różnych klientów oraz nie była zbyt łatwa do zbruteforcowania.

- **Pieprz/pepper/secret key (NIST tak to nazywa)** – Pieprz powinien być przechowywany w osobnej bazie danych (nie razem z solą i haszem) albo wcale nie być przechowywany.
    - Żeby atakujący musiał bruteforcować się przez niego. Pieprz powinien mieć według NIST co najmniej 112 bitów, aby jego bruteforce nie był łatwy.
- **Tęczowe tablice** – słowniki/bazy danych z policzonymi hashami dla potencjalnych haseł, przyspieszające proces łamania hasła. Np. w Wi–Fi SSID sieci jest używany jako sól do haszu, więc znając SSID, możemy przygotować tęczową tablicę do łamania PSK.
    - Rainbow tables are a practical example of a space–time tradeoff: they use less computer processing time and more storage than a brute–force attack which calculates a hash on every attempt, but more processing time and less storage than a simple table that stores the hash of every possible password.

Wymagania związane z bezpieczeństwem kluczy kryptograficznych i zarządzanie kluczami

Poufność kluczy

Integralność kluczy

Bezpieczne generowanie – losowo, odpowiednia długość (AES – 256 min) RSA (2048 min.)

Okres ważności kluczy

Klucze przesyłane np. z użyciem TPM – w bezpieczny sposób

na podstawie m.in.: GDPR (General Data Protection Regulation), czyli RODO – dla ochrony danych osobowych w UE, PCI DSS (payment card industry data security standard), NIST SP 800–57

Hardware security module (HSM) – urządzenie fizyczne, które bezpiecznie generuje, przechowuje i zarządza kluczami szyfrującymi. np. AWS CloudHSM

- Tak samo TPM, KMS (key management systems)

Przechowywanie w RAM tylko na czas używania

- Usuwanie np. przez shred w Linuxie – nadpisuje losowymi znakami, więc ciężko odzyskać

Zarządzanie kluczami musi być oddzielone od operacji, w których te klucze są używane.

Dystrybucja/ustalanie kluczy kryptograficznych:

- Wymiana typu punkt–punkt
- Centrum Dystrybucji Kluczy
- Algorytm Shamira i Diffie–Hellmana
    - Algorytm Shamira (Three–pass protocol) – analogia do skrzyni z dwoma kłódkami

    - Algorytm Diffie–Hellmana – analogia do mieszania kolorów (jeden znany kolor i kolory tajne jako klucze prywatne). Wymieszany kolor jest używany jako sekret.
- Uczenie maszynowe

Zarządzanie:

generowanie–> rejestracja –> przechowywanie –> dystrybucja –> używanie –> wymiana na nowy (rotation – co 365 dni) –> archiwizacja –> niszczenie

Dla dystrybucji: TLS, IPsec, SSH – przesyłanie tylko zaszyfrowanych

Infrastruktura klucza publicznego (Public Key Infrastructure, PKI) – zespół urządzeń, oprogramowania, procedur, polityk i osób umożliwiający bezpieczne używanie kluczy publicznych w formie cyfrowych certyfikatów.

- Urząd rejestracji – weryfikacja danych i rejestracja (RA)
- Urząd certyfikacji – wydawanie certyfikatów (podpisanych kluczem prywatnym CA)
- Repozytoria – unieważnione certyfikaty – Certificate revocation list (CRL)

PKI umożliwia tworzenie, przechowywanie, zarządzanie i rozprowadzanie certyfikatów klucza publicznego

Nie zapewnia zaufania do jednostki (zapewnia jedynie zaufanie do klucza tej jednostki)

OCSP (Online Certificate Status Protocol) jest protokołem umożliwiającym sprawdzanie czy weryfikowany certyfikat jest ważny.

- Prosty protokół typu żądanie–odpowiedź
- Wymaga zaufanej jednostki zwanej respondentem OCSP, która odsyła odpowiedzi podpisane cyfrowo
- Informuje tylko, czy certyfikat został unieważniony

Rozpowszechnianie prywatne – użytkownicy przesyłają innym swoje certyfikaty (np. PGP)

Repozytoria – certyfikaty są publikowane w ogólnodostępnym miejscu (np. zdalny serwer LDAP, serwer FTP, serwer www + rozpoznawalny adres DNS, itp.)

Analiza malware

Typy oprogramowania złośliwego i sposoby działania oraz ich wykrywanie i analiza

1) **Wirusy** – programy, które dołączają się do innego programu, żeby się powielać,
2) **Robaki** – podobnie jak wirusy, ale rozprzestrzeniają się przez sieć,
3) **Trojany** – podszywają się pod legit oprogramowanie, zwykle działają jako keyloggery, boty lub ransomware,
4) **Ransomware** – oprogramowanie wymuszające okup, WannaCry
5) **Rootkity** – ukrywają złośliwe oprogramowanie w systemie, zapewniając dostęp do systemu,

6) **Boty** – do DDoSów, kopania krypto,
7) **Adware** – wiadomo,
8) **Exploity** – wykorzystują luki w legit oprogramowaniu, żeby zainstalować malware albo przejąć kontrolę.

Wykrywanie:

- **Antywirusy** – skanują pliki, porównują sygnatury i heurystyki,
- **Firewalle** – blokują niepożądany ruch sieciowy,
- **IDS/IPS** – j.w., analiza i blokowanie na poziomie sieci,
- **Sandboxing** – analiza w sandboxie, dynamicznie co jest robione w systemie, jakie połączenia nawiązuje, co modyfikuje,

Analiza:

- **Statyczna** – badanie kodu lub binarki
- **Dynamiczna** – uruchamianie malware w kontrolowanym środowisku (sandbox j.w.), debugowanie

Inżynieria wsteczna oprogramowania i techniki ukrywania złośliwego oprogramowania (obfuskacja, pakowanie, szyfrowanie, itp.)

Narzędzia:

- **IDA** – dekompilacja binarek
- **Ghidra** – dekompilacja binarek
- **OllyDbg** – debugger
- **x64dbg** – debugger dla Windowsa
- **Radare2** – analiza statyczna i dynamiczna

Elementy analizy:

- Identyfikacja pliku
- Analiza statyczna – stringi, funkcje, struktury plików
- Analiza dynamiczna – monitorowanie rejestrów, plików, ruchu sieciowego
- Odwracanie kodu – dekompilacja/deobfuskacja
- Ostatecznie chcemy stworzyć reguły, które pozwolą wykrywać malware

Ukrywanie malware:

- **Obfuskacja** – zaciemnianie kodu źródłowego (nieczytelne nazwy, nieużywane instrukcje, etc.)
- **Pakowanie** – kompresja/kodowanie pliku w celu ukrycia zawartości (jak znajdziemy typ packera to możemy rozpakować)
- **Szyfrowanie** – jakieś części kodu/plików są szyfrowane, przez co nie wykryjemy malware po sygnaturach, klucz może być albo gdzieś w kodzie, albo pobierany z zewnętrznego serwera.
- **Różne techniki utrudniające debugowanie**
    - Wykrywanie maszyn wirtualnych, debuggerów, sandboxów,
    - Wykonywanie się przy jakichś predefiniowanych warunkach
        - np. dla konkretnego typu systemu, po konkretnej dacie itd.

- **Wstrzykiwanie kodu** – malware wstrzykuje swój kod do pamięci innego legit procesu i wykonuje się w ramach niego, przez co trudno go wykryć (DLL injection, process hollowing, itp.)"

Informatyka śledcza i wykrywanie anomalii

Informatyka śledcza (ang. Digital Forensics)

Jest to dziedzina zajmująca się identyfikacją, analizą i zabezpieczaniem dowodów cyfrowych. Główne obszary to:

- **Analiza systemów komputerowych**: Odzyskiwanie i analiza danych z komputerów, dysków twardych, urządzeń przenośnych itp.
- **Śledztwa sieciowe**: Analiza ruchu sieciowego w celu identyfikacji incydentów, takich jak włamania, ataki DDoS czy wycieki danych.
- **Analiza złośliwego oprogramowania**: Badanie plików i programów pod kątem ich szkodliwego działania.
- **Zabezpieczanie dowodów**: Przygotowywanie dowodów cyfrowych w sposób akceptowalny w procesach prawnych.

Wykrywanie anomalii

Jest to proces identyfikacji nietypowych wzorców w danych, które mogą wskazywać na:

- Próby naruszenia bezpieczeństwa (np. ataki sieciowe).
- Błędy systemowe.
- Nieautoryzowane działania użytkowników.

Metody wykrywania anomalii

1. **Oparte na regułach**:
   - Predefiniowane reguły (np. firewalle, systemy IDS/IPS).
   - Rozpoznawanie podejrzanych działań na podstawie znanych wzorców.
2. **Oparte na danych**:
   - **Metody statystyczne**: Analiza odchyleń od normy (np. analiza ruchu sieciowego pod kątem nadmiernej liczby pakietów).
   - **Uczenie maszynowe**:
     1. **Algorytmy klasyfikacyjne**: Modele takie jak SVM, drzewa decyzyjne czy sieci neuronowe mogą klasyfikować dane jako normalne lub anomalne.
     2. **Klasteryzacja**: Algorytmy takie jak K-means czy DBSCAN grupują dane, identyfikując odstające przypadki.

Zastosowanie wykrywania anomalii w informatyce śledczej

1. **Analiza ruchu sieciowego**:
   - Wykrywanie nietypowych wzorców ruchu wskazujących na ataki DDoS, skanowanie portów czy próby włamań.
   - Identyfikacja nietypowych zachowań użytkowników (np. nieautoryzowane logowania).
2. **Odzyskiwanie danych**:
   - Identyfikacja plików zmodyfikowanych przez złośliwe oprogramowanie.

    - Wykrywanie ukrytych danych (np. w steganografii).
3. **Analiza logów**:
    - Detekcja anomalii w logach systemowych, takich jak nietypowe logowania, błędy czy zmiany konfiguracji.
4. **Badanie złośliwego oprogramowania**:
    - Analiza behawioralna programów w celu wykrycia nietypowych działań (np. nieautoryzowanego dostępu do systemów plików).

Bazy danych

Porównanie relacyjnych baz danych z nierelacyjnymi (NoSQL)

Relacyjna baza danych lub **relacyjny system zarządzania bazą danych (RDMS)** przechowuje informacje w **tabelach**. Często te tabele mają **wspólne informacje** między sobą, co powoduje utworzenie **relacji** między tabelami.

Tabela używa **kolumn** do definiowania przechowywanych informacji i **wierszy** dla danych. Każda tabela ma kolumnę, która musi mieć **unikalne wartości** — **klucz podstawowy**.

Kolumna ta, kiedy używana w innych tabelach (dla zdefiniowania relacje między nimi) to **klucz obcy**.

Przez SQL można wchodzić z bazą w interakcje.

Jeśli dane są przewidywalne pod względem struktury, rozmiaru i częstotliwości dostępu, relacyjne bazy danych są najlepszym wyborem.

Plusy:

- ACID (Atomicity, Consistency, Isolation, Durability)
- Normalizacja – polega na zapewnieniu, że dane są zorganizowane w taki sposób, aby anomalie danych zostały zredukowane lub wyeliminowane. To z kolei zmniejsza koszty przechowywania.
- Prostota

Minusy:

- Brak skalowalności
- Nieelastyczność – sztywny schemat, format, długość – zdefiniowane z góry
- Wydajność – Wraz ze wzrostem złożoności wydłuża się również czas wykonywania zapytań.

Przykłady: MySQL, PostgreSQL, Oracle, MS SQL Server

Nierelacyjna baza danych, czasami nazywana **NoSQL (Not Only SQL)**, to dowolny rodzaj bazy danych, który **nie używa tabel, pól i kolumn**, czyli koncepcji danych strukturalnych z relacyjnych baz danych.

Typy baz danych (ze względu na sposób przechowywania danych):

- **Bazy danych dokumentów** przechowują dane w dokumentach,
    - Zazwyczaj struktury typu JSON, obsługujące różne typy danych.

- **Klucz–wartość**
- **Graf**

Plusy:

- Skalowalność pozioma
- Elastyczność schematu
- Wydajność przy dużych zbiorach danych

Minusy:

- Brak ACID
- Brak standaryzacji

Jeśli przechowywane dane **muszą być elastyczne pod względem kształtu lub rozmiaru** lub **muszą być otwarte na zmiany w przyszłości**, odpowiedzią jest **nierelacyjna baza danych**.

Przykłady: MongoDB, Cassandra, Redis, Couchbase, Neo4j

Transakcja bazodanowa i jej wymagane cechy

Transakcje w bazach danych zapewniają, że operacje są wykonywane w sposób **spójny, niezawodny i bezpieczny**, **nawet w przypadku błędów systemowych, awarii czy innych problemów**.

W kontekście baz danych, transakcja to **sekwencja operacji** (np. **dodawanie, modyfikowanie, usuwanie danych**), która jest **traktowana jako jedna całość**.

Wymagane cechy transakcji bazodanowej (ACID):

Transakcja bazodanowa powinna spełniać zestaw właściwości określanych jako ACID:

1) Atomicity (Niepodzielność)

- Transakcja jest niepodzielna i nieprzerwana.
  - Jeśli jedna część transakcji się nie powiedzie, cała transakcja jest cofana (rollback).
- **Przykład** – W przelewie bankowym kwota musi zostać odjęta z jednego konta i dodana do drugiego. Jeśli operacja na jednym koncie się nie powiedzie, zmiany na obu kontach zostaną anulowane.

2) Consistency (Spójność)

- Transakcja przenosi bazę danych z jednego stanu spójnego do innego.
- Reguły integralności (np. klucze obce, ograniczenia unikalności) muszą być zachowane.
- **Przykład** – Jeśli przelewasz pieniądze, saldo końcowe obu kont (nadawcy i odbiorcy) musi być zgodne z regułami rachunkowości.

3) Isolation (Izolacja)

- Równoczesne transakcje nie powinny wpływać na siebie nawzajem.
  - Każda transakcja jest izolowana od innych, dopóki nie zostanie zatwierdzona.

- **Przykład** – Dwa równoczesne przelewy na to samo konto nie mogą prowadzić do sytuacji, w której końcowe saldo jest nieprawidłowe.

4) Durability (Trwałość)

- Po zatwierdzeniu transakcji (commit), jej efekty są trwałe i zapisywane na nośniku pamięci niezależnym od awarii (np. dysku).
- **Przykład** – Po zapisaniu przelewu na koncie, zmiana jest zachowana nawet w przypadku awarii systemu.

Etapy transakcji:

1) Start transakcji:

- Początek operacji, które będą częścią transakcji.

2) Wykonanie operacji:

- Wykonywanie zapytań i modyfikacji na danych.

3) Commit (zatwierdzenie):

- Zatwierdzenie transakcji, zmiany są trwałe.

4) Rollback (wycofanie):

- Cofnięcie wszystkich operacji, jeśli wystąpi błąd.

Język SQL – cechy języka, problem interpretacji NULL, zagadnienie kolejności wierszy i kolumn w tabeli

Język SQL – jest to **język programowania**, który określa się mianem **języka dziedzinowego** – takiego, który został **stworzony pod konkretną dziedzinę** i **służy do określonego celu**.

Pozwala na operowanie na bazach danych w tym:

- Przechowywanie danych
- Aktualizacja danych
- Usuwanie danych
- Wyszukiwanie danych

SQL jest **językiem deklaratywnym** – użytkownik określa co chce uzyskać, a nie w jaki sposób. Sam język jest opisany w standardzie ANSI/ISO,

- Poszczególna składnia funkcji czy sposób ich wykonywanie może się różnić w poszczególnych systemach bazodanowych.

Sam język dzieli się na kilka poszczególnych podjęzyków, które mają swoje zadania takiej jak tworzenie bazy danych, wyszukiwanie samych danych itd.

SQL używa się raczej w celu ,,dodatku'' do aplikacji, nie stosuje się go do tworzenia samych aplikacji.

NULL – „brak” wartości w danej komórce w bazie danych.

- Jeśli dana wartość jest opcjonalna w komórce w bazie danych i nie wpiszemy konkretnej wartości, uzyska ona właśnie wartość NULL.

- Nie jest to ani pusty string, ani zero. Raczej jest to wartość „**niewiadomo”** lub **jej nie ma**.
- NULL nie można porównywać operatorami takimi jak =, <, > czy <>,
    - **Żadne** dwie wartości NULL **nie są sobie równe**.
- Używa się wyrażeń **IS NULL** i **IS NOT NULL**, aby sprawdzić, czy wartość w komórce to NULL.
- Wartość NULL możemy wsadzić w dowolną komórkę o dowolnym typie danych.
    - Boolean, który zwraca to UNKNOWN.

W bazie danych **nie mamy naturalnej kolejności wierszy** – jest to **zbiór nieuporządkowany** i to **my określamy** w jaki sposób te dane mają zostać uporządkowane.

Porządkować dane można choćby dzięki klauzuli ORDER BY. Podajemy tam kolejne kolumny jako kolejność sortowania, które różni się w zależności od typu danych w kolumnie. Bez **ORDER BY** nie mamy pewności co do uporządkowania wyświetlania danych.

Kolejność kolumn określamy przy tworzeniu tabeli w bazie danych. Przy otrzymywaniu danych z bazy kolumny możemy w dowolny sposób wyświetlać w określonej przez nas kolejności, a także w dowolnej kolejności na nich operować (tzn. używać słów **JOIN** itd.).

Każda kolumna ma swój typ danych, przy czym można określić, czy mogą w niej się znaleźć wartości NULL – w takim przypadku przy dodawaniu nowego nie jest wymagane podawanie wartości dla tej kolumny.

dTypowe słowa: SELECT, FROM, WHERE, INSERT, UPDATE, DELETE, CREATE, ALTER, DROP, JOIN, GROUP BY, ORDER BY, HAVING, DISTINCT, LIMIT, AND, OR, NOT, IN, BETWEEN, LIKE, NULL

Bezpieczeństwo bezprzewodowych sieci komputerowych

Podstawy transmisji bezprzewodowej oraz metody wielodostępu do kanału radiowego w sieci Wi–Fi

Transmisja bezprzewodowa polega na **przesyłaniu danych za pomocą fal radiowych** w przestrzeni otwartej, **bez użycia fizycznego medium transmisyjnego**, jak przewody.

Metody transmisji:

1) FHSS (Frequency Hopping Spread Spectrum):

- **Zasada działania**: dla sieci istnieje dobrze znana sekwencja przeskakiwania, która mówi, do którego kanału system przeskakuje przy następnym przeskakiwaniu.
- W Wi–Fi mało używany, początkowo znalazł zastosowanie do sterowania rakiet, bo ciężej go zakłócić, kiedy skaczemy po częstotliwościach w czasie.
    - Słabo pracuje dla dużych szybkości

- Nadajnik działa **na jednym kanale na raz**
- **Zalety**: prosta implementacja, używa tylko niewielkiej części widma w dowolnym momencie
- **Wady**: nie jest tak solidny jak DSSS, łatwiejszy do wykrycia

2) DSSS (Direct Sequence Spread Spectrum):

- **Zasada działania**: Rozpraszamy sygnał danych na szerokie pasmo przy użyciu unikalnego kodu pseudolosowego (PN).
    - Operacja XOR sygnału i PN
- Odbiorca używa **tego samego kodu do odzyskania sygnału**, co **zwiększa odporność na zakłócenia** i **podsłuch**.
- **Zaleta**: zmniejsza selektywne zanikanie częstotliwości
- DSSS ma **najmniejszy overhead** (IR już niestosowane), dlatego opłaca się jego a nie FHSS

3) IR (Infrared):

- **Zasada działania**: podczerwień jak w pilocie
- **Wada/zaleta**: sygnał jest blokowany przez ściany

4) OFDM (Orthogonal Frequency Division Multiplexing):

- **Zasada działania**: Dzielimy pasmo na wiele wąskopasmowych, ortogonalnych podnośnych, a dane są transmitowane równolegle na tych podnośnych.

W sieciach Wi–Fi stosowano głównie DSSS i OFMD. Poza zasadą działania techniki wielodostępu różnią się między sobą takimi parametrami jak **nagłówek warstwy fizycznej**, **czas trwania szczelin czasowych** oraz **modulacja**.

W zależności od standardu Wi–Fi transmisja bezprzewodowa odbywa się z różnymi parametrami takimi jak:

- **Pasmo częstotliwości** (2.4 GHz, 5GHz, 6GHz)
- **Szerokość kanału** (np. 20MHz, 80 MHz, w nowszych standardach tendencja do szerszych kanałów)
- **Modulacja** (np.
    - **BPSK (Binary Phase Shift Keying)** – koduje 1 bit na cykl
    - **QPSK (Quadrature Phase Shift Keying)** – koduje 2 bity na cykl
    - **QAM (Quadrature Amplitude Modulation)** (im wyższe tym lepsze, ale wyższe wymagania dotyczące jakości sygnału i odbiornika):
        - 16–QAM,
        - 64–QAM,
        - 256–QAM,
        - 1024–QAM

Metody wielodostępu:

Kiedy już umiemy przesyłać informacje w falach radiowych, trzeba jakoś ustalić, kto, kiedy ma wysyłać, a kto słuchać. W sieciach bezprzewodowych wykrywanie kolizji jest bardzo trudne lub wręcz niemożliwe, bo **moc sygnału wraz z odległością maleje**.

Jeśli na tym samym obwodzie antenowym (ta sama antena do odbioru i nadawania), wysyłamy sygnał o mocy 100mW i odbieramy jednocześnie ze zmniejszoną mocą (bo zmalała wraz z odległością, którą przebyła), to **nie jesteśmy w stanie wykryć**, że coś przyszło, kiedy my jednocześnie nadajemy na antenie.

Dlatego staramy się **unikać kolizji** (które i tak będą się zdarzać) zamiast je wykrywać.

CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)

- **Nasłuchiwanie kanału**: Przed rozpoczęciem transmisji urządzenie sprawdza, czy kanał komunikacyjny jest wolny
    - Żadne inne urządzenie nie nadaje w tym momencie.
- **Unikanie kolizji**: Jeśli kanał jest zajęty, urządzenie wstrzymuje się z nadawaniem. Jeśli jest wolny, urządzenie przechodzi do kroku nadawania.
- **Weryfikacja i potwierdzenie**: Po wysłaniu danych odbiorca może potwierdzić odbiór
    - np. poprzez ACK.

Mechanizm backoff

- **Element CSMA/CA** – zapewnia **uczciwy podział kanału**, przynajmniej **pod względem ilości transmisji**. Polega na tym, że losujemy **backoff**, który odczekujemy przed nadaniem ramki (chyba że to pierwsza ramka to wystarczy odczekać czas DIFS). Poza tym:
    - Jeśli nadamy ramkę i otrzymamy ACK to **$CW = CW_{min}$**.
    - Jeśli nadamy ramkę i NIE otrzymamy ACK to "podwajamy" wartość **CW=(2*CW+1)**

Backoff

- Ilość szczelin slot time z zakresu od (0, CW), gdzie **CW (Contention Window)** należy do przedziału (**CW_{min}**, **CW_{max}**) – dla WiFi to np. (7, 1023).

Mechanizm RTS/CTS

- Polega na tym, że przed wysłaniem długiej ramki z danymi, wysyłamy ramkę RTS (Request to Send) i czekamy na odpowiedź – CTS (Clear to Send).
- Rozwiązuje to problem **hidden nodes**, który polega na tym, że jak mamy AP między dwiema stacjami, to, gdy $Client_1$ nadaje do AP, $Client_2$ może tego nie "słyszeć", zacznie dawać w tym samym czasie i przeszkodzi w transmisji.

$Client_1$ AP $Client_2$

Mechanizmy bezpieczeństwa stosowane w sieciach Wi–Fi (np. WPA) i standardzie Bluetooth

1) WiFi (IEEE 802.11)

WEP

- Opisany w kolejnym zagadnieniu

WPA (WiFi Protected Access)

- Ulepszony **RC4**,
- Wykorzystuje **TKIP (Temporal Key Integrity Protocol)** jako mechanizm szyfrowania.
    - TKIP dynamicznie zmienia klucze szyfrowania, co zwiększa bezpieczeństwo (ale wydajność spada),
- **Zapewnia ochronę przed częścią ataków na WEP**, takich jak łatwe łamanie statycznych kluczy,
- **Kompatybilny ze starszym sprzętem** (wymagał tylko aktualizacji oprogramowania).

WPA2

- Wprowadza bardziej zaawansowany protokół szyfrowania **AES jako standard**.
- Zastępuje TKIP bardziej bezpiecznym i nowoczesnym mechanizmem szyfrowania.
- Wykorzystuje **CCMP (Counter Mode Cipher Block Chaining Message Authentication Code Protocol)** jako protokół zarządzania integralnością danych.
- **Obsługuje dwa tryby**:
    - **Personal Mode (PSK)**: Dla domowych użytkowników, oparty na wspólnym kluczu.
    - **Enterprise Mode**: Dla sieci korporacyjnych, wymaga serwera uwierzytelniania (RADIUS).

802.11w (Protected Management Frames (PMF))

- 802.11w jest **opcjonalnym dodatkiem do WPA2** i **obowiązkowym do WPA3**,
- **Każda chroniona ramka zarządzania** jest **uwierzytelniana**, co zapewnia, że pochodzi od prawidłowego źródła
    - Zapobiega atakom np. Deauthentication, Dissassociation Flooding

WPA3

- Wprowadza **SAE (Simultaneous Authentication of Equals)**, znacznie bezpieczniejszy mechanizm wymiany kluczy, który zastępuje PSK.
- **Dodatkowe funkcje**:
    - **Forward Secrecy**: Zapewnia, że nawet jeśli klucz zostanie złamany, wcześniejsze dane pozostają bezpieczne.
    - **Ochrona publicznych sieci Wi–Fi**: Szyfrowanie indywidualnych połączeń, nawet w otwartych sieciach (opcja **OWE – Opportunistic Wireless Encryption**).

2) Bluetooth (IEEE 802.15 WPAN (Wireless Personal Area Networks) a dokładnie 802.15.1)

Podstawowe komponenty bezpieczeństwa Bluetooth:

- **Adres urządzenia Bluetooth (BD_ADDR)**: unikalny, 48–bitowy adres przypisany każdemu urządzeniu Bluetooth przez IEEE.
- **Klucz autoryzacji**: 128–bitowy losowy numer używany do uwierzytelniania.

- **Klucz szyfrowania**: długość od 8 do 128 bitów, wykorzystywany do szyfrowania danych.
- **Liczba losowa (RAND)**: zmienny 128–bitowy numer generowany przez urządzenie.

Tryby bezpieczeństwa Bluetooth:

- **Tryb 1**: Brak zabezpieczeń.
- **Tryb 2**: Zabezpieczenia wymuszane na poziomie usługi.
- **Tryb 3**: Zabezpieczenia wymuszane na poziomie linku przed ustanowieniem kanału.
- **Tryb 4**: Zabezpieczenia na poziomie linku z wymianą klucza szyfrowanego.

Zarządzanie kluczami – Link Key:

- 128–bitowy losowy numer.
- Służy do uwierzytelniania urządzeń i jako parametr do generowania klucza szyfrowania.

Żywotność:

- **Semi–permanent (półtrwały)**:
  - Przechowywany w pamięci nieulotnej.
  - Może być używany ponownie w kolejnych sesjach do uwierzytelniania urządzeń, które go współdzielą.
- **Temporary (tymczasowy)**:
  - Obowiązuje tylko w bieżącej sesji.
  - Po zakończeniu sesji nie może być ponownie wykorzystany.

Rodzaje Link Keys:

- **Klucz inicjalizacyjny**: używany podczas konfiguracji urządzenia.
- **Klucz jednostkowy**: przypisany do urządzenia, zmieniany rzadko.
- **Klucz kombinacyjny**: generowany dla każdej pary urządzeń.
- **Klucz główny**: tymczasowy, stosowany przy transmisji do wielu odbiorców.
- **Klucz szyfrowania**: 128–bitowy, generowany na podstawie klucza linku.

Szyfrowanie transmisji:

- od BT 4.0 **AES–128 w trybie CCM** (dokładnie to samo co w WiFi)

Ulepszone mechanizmy parowania w Bluetooth 2.1:

- **Legacy Pairing**: wprowadzanie kodu PIN na obu urządzeniach.
- **Secure Simple Pairing (SSP)**:
  - **Just Works**: brak interakcji użytkownika, ale niższe zabezpieczenia przed atakami MITM.
  - **Numeric Comparison**: porównanie wyświetlanych kodów.
  - **Passkey Entry**: wprowadzanie kodu przez użytkownika.
  - **Out of Band (OOB)**: wymiana danych za pomocą zewnętrznego medium, np. NFC

Podatności i wady mechanizmów WEP i WPS oraz ataki na sieci Wi–Fi

1) WEP (Wired Equivalent Privacy)

Standard zabezpieczeń **stosowany przed wprowadzeniem WPA/WPA2 w rozszerzeniu 802.11i**. Z wielu powodów był podatny i zrobiony w nieprzemyślany sposób.

Mechanizmy zabezpieczeń, które wprowadza WEP:

- **Authentication** (aby kontrolować dostęp do sieci)
- **Encryption** (aby zapobiec podsłuchiwaniu)

Wady mechanizmu WEP:

- Korzysta z algorytmu strumieniowego **RC4**
- Ma **statyczny 40–bitowy base key** (64–bitowy klucz na pakiet i 24–bitowy IV)
- Używa **Integrity Check Value (ICV)** – **CRC–32**
- **Długość IV (24–bity)** to zdecydowanie za mało, przez co szanse na zduplikowane IV są bardzo duże
- WEP używa **tych samych kluczy** do **Authentication** i **Encryption**
- **Nie da się** zarządzać kluczami
- **Ten sam klucz statyczny używany na AP i wszystkich klientach**
- Szanse na odzyskanie 128–bitowego klucza są bardzo wysokie po przechwyceniu tysięcy ramek
- **Słabe IV** (klucze, dla których wzór bitów w pierwszych 3 bajtach klucza powoduje pojawienie się podobnych wzorów w pierwszych bajtach sekwencji szyfrującej)

2) WPS (Wi–Fi Protected Setup)

Ułatwia podłączenie urządzenia do sieci bezprzewodowej z zabezpieczeniami.

Korzysta z **algorytmu Diffiego Hellmana**, działa w ramach sesji EAP

Znane też pod innymi nazwami:

- Wi–Fi Simple Config,
- Push 'n' Connect,
- PBC,
- Quick Secure Setup (QSS)

Działa w następującej kolejności:

- **Wymiana kluczy** Diffiego Hellmana,
- Sprawdzenie **pierwszej części pinu**,
- Sprawdzenie **drugiej części pinu**,
- **Decyzja** o przyznaniu dostępu.

Możliwe tryby działania:

- **Obowiązkowe**:
  - PIN
  - PBC

- **Opcjonalne**:
    - NFC
    - USB

Tryb PIN szczegóły (jednocześnie wady):

- **Podatny na bruteforce**,
- **Zaledwie 8 cyfr**, sprawdzanych po 4,
- W trybie **WPS Sticker PIN** ostatnia cyfra jest **sumą kontrolną** (11 000 kombinacji),
- W trybie **WPS User–Defined** – PIN – $10^4 + 10^4 = 20\,000$ kombinacji,
- Później dodano blokadę na 60 s po 3 nieudanych próbach (**łamanie zajmie 3 dni**).

Tryb PBC (push button) szczegóły:

- Ustawia PIN dostępowy na 00000000 (ustandaryzowane) i stacja próbuje tego PINu i uzyskuje dostęp do sieci

Pozostałe ataki na sieci WiFi:

- **Constant Jamming** – nakurwiasz cały czas jakimś syfem w kanał i nikt nic nie może wysłać wtedy, bo kanał ciągle zajęty,
- **Reactive Jamming** – nakurwiasz precyzyjnie, jak Kapitan Bomba z laserka do kurwinoxa, w ramki, które chcesz zepsuć, i psujesz np. ramki w beacon i klient posrany jak sułtan kosmitów, bo nie może sieci WiFi znaleźć,
    - Keep in mind, że ze względu na przestrzenną naturę fal, Reactive Jamming nie działa tak dobrze jak Constant, bo jeśli atakujący jest dalej od AP niż ty, to jest szansa, że zdążysz odebrać ramkę, zanim atakujący ją uszkodzi
- **CW Min/Max Cheating/Disable Virtual Carrier Sense/ Ignore Backoff** – tak jak zostało opisane powyżej, stacje powinny losować, ile szczelin czasowych mają odczekać przed nadaniem ramki, ale mogą oszukiwać i losować mniej albo w ogóle ignorować ustalone zasady
- **Forcing corrupt FCS** – taki trochę nie do końca atak; zmuszamy kartę, **żeby liczyła błędny FCS** i wysyłamy ramki, przez co jakiś system monitorujący sieć myśli, że w sieci jest dużo uszkodzonych ramek
- **Fake Power Save** – polega na wysyłaniu **fałszywych ramek sygnalizujących tryb oszczędzania energii**, co może zakłócić komunikację między klientem a AP
- **Deauthentication and Disassociation Flooding** – flooding tytułowych ramek, powodujący rozłączenie klientów z siecią
    - zmianę stanu na **unauthorized** lub **disassociated**
- **Probe Response Flooding** – polega na wysyłaniu **fałszywych ramek Probe Response** (z losowymi SSID sieci) **na zapytania Probe Request**, przez co klient widzi masę nieistniejących sieci
- **Reassociation Request Flooding** – ma na celu przeciążenie AP
- **Beacon Flooding** – polega na wysyłaniu ogromnej liczby **fałszywych ramek Beacon**, tworząc widmo **nieistniejących sieci WiFi**, co wprowadza zamieszanie wśród klientów i utrudnia wybór rzeczywistej sieci (podobnie jak w Probe Response Flooding)
- **Evil Twin** – polega na utworzeniu **fałszywego punktu dostępowego imitującego prawdziwą sieć** (ten sam adres MAC AP (BSSID) i SSID)

- **Rogue AP** – **nieautoryzowany AP**, np. pracownik tworzy hotspot ze słabym zabezpieczeniem/bez zabezpieczenia i stwarza zagrożenie
- **Honeypot** – sieć WiFi stworzona w celu przyciągnięcia użytkowników i monitorowania ich aktywności lub przechwytywania danych.
- **Dictionary Attack** – przechwytujemy 4–way–handshake jakiejkolwiek stacji i próbujemy złamać hasło

Bezpieczeństwo oprogramowania

Wybór języka programowania a bezpieczeństwo tworzonej przy jego pomocy aplikacji

1) Typowanie danych:

Języki statycznie typowane:

Wykrywają błędy związane z typami danych w czasie kompilacji.

- **Przykłady**: C++, Java, Rust, Go.
- **Zaleta**: Zapobiegają wielu błędom, np. przekazywaniu nieodpowiednich danych do funkcji.
- **Wada**: Mogą być mniej elastyczne w szybkich projektach.

Języki dynamicznie typowane:

Sprawdzają typy danych w czasie wykonywania.

- **Przykłady**: Python, JavaScript, Ruby.
- **Zaleta**: Szybsze prototypowanie.
- **Wada**: Większe ryzyko błędów typu `TypeError` w czasie wykonywania.

2) Zarządzanie pamięcią:

Ręczne zarządzanie pamięcią:

Języki takie jak **C** i **C++** wymagają jawnego zarządzania pamięcią.

- **Ryzyko**: Błędy, takie jak przepełnienie bufora, wycieki pamięci, dereferencja wskaźnika zerowego.

Automatyczne zarządzanie pamięcią (Garbage Collection):

Języki takie jak **Java**, **Python**, **C#**, **Go**.

- **Zaleta**: Eliminuje ryzyko wycieków pamięci.
- **Wada**: Może wprowadzać opóźnienia związane z działaniem mechanizmu Garbage Collection.

Bezpieczeństwo na poziomie języka:

Języki takie jak **Rust** zapewniają bezpieczne zarządzanie pamięcią bez garbage collection, dzięki mechanizmom typu `**ownership**` i `**borrowing**`.

3) Obsługa wyjątków

Mechanizmy obsługi wyjątków:

Języki takie jak **Java** czy **Python** oferują zaawansowane mechanizmy obsługi wyjątków.

- **Zaleta**: Minimalizacja ryzyka błędów podczas obsługi nieoczekiwanych sytuacji.
- **Ryzyko**: W niektórych językach (np. C++) brak obsługi wyjątków może powodować niekontrolowane awarie.

4) Dostępność narzędzi i bibliotek bezpieczeństwa:

Języki z rozwiniętym ekosystemem (np. **Python**, **Java**, **JavaScript**) oferują liczne biblioteki wspierające bezpieczeństwo, takie jak:

- Szyfrowanie (OpenSSL, pycryptodome).
- Weryfikacja danych wejściowych (Express Validator, Joi).

5. Łatwość analizy kodu:

Języki prostsze i bardziej czytelne (np. **Python**, **Go**) są łatwiejsze do audytu pod kątem bezpieczeństwa niż bardziej złożone (np. **C++).**

Bezpieczeństwo aplikacji internetowych i mobilnych – sposoby ochrony, podatności, ataki, ocena bezpieczeństwa

1) Internetowe:

Sposoby ochrony:

- **SAST** (static) i **DAST** (dynamic app security testing),
- **Wyłączanie zbędnych funkcji i usług**,
- **WAF (Web Application Firewall)** – filtrowanie i monitorowanie ruchu HTTP między aplikacją a Internetem. Zazwyczaj chroni przed np. cross–site forgery, XSS, SQL injection.
    - To obrona warstwy protokołu 7 (w modelu OSI) i nie broni przed wszystkim. Ta metoda łagodzenia ataków jest zwykle częścią zestawu narzędzi, które razem tworzą holistyczną obronę przed szeregiem wektorów ataku.
- **DNSSEC** – Protokół gwarantujący, że ruch DNS aplikacji internetowej jest bezpiecznie kierowany do właściwych serwerów (do MITM),
- **Certyfikaty, HSTS (HTTP Strict Transport Security)** – wymuszenie HTTPS w przeglądarce, MFA (uwierzytelnienie wieloskładnikowe), patchowanie systemów,
- **CSP (Content–Security–Policy)** – pozwala twórcy aplikacji webowej na ścisłe zdefiniowanie, skąd mogą pochodzić dodatkowe zasoby, z których korzysta aplikacja www (.js, css) – takie whitelisty tworzy (na XSS) (czyli co można załadować i wykonać na stronie),
- **CORS (Cross–Origin Resource Sharing)** – mechanizm, który pozwala kontrolować, jakie domeny mogą wykonywać żądania do API lub serwera, np. fetch() (na CSRF) (czyli kto ma dostęp do zasobów serwera),

Parametry ciasteczek, które warto ustawić:

- **Path** – określa ścieżkę URL, na której ciasteczko jest dostępne

- **Expires** – określa datę i godzinę wygaśnięcia ciasteczka
- **Secure** – wskazuje, że ciasteczko powinno być wysyłane tylko przez HTTPS
- **HttpOnly** – zapobiega dostępowi do ciasteczka przez język JS. Ciasteczko jest dostępne tylko przez protokół HTTP lub HTTPS (chroni przed XSS)

Podatności i ataki: XSS, Broken Access Control, Injection, Insecure Design, SSRF, CSRF, IDOR, Cryptographic failures, DOS, XXE (XML External Entity), słabe hasła, deprecated haszowanie, nieunieważnienie sesji po wylogowaniu, Clickjacking (ukrycie pod teoretycznie bezpiecznym i godnym zaufania elementem strony innej zawartości), MITM (on–path), session sniffing,

Ocena bezpieczeństwa: przeprowadzenie testów penetracyjnych, threat modeling (badanie potencjalnych wektorów ataku i sposobów ich mitygacji), bug bounty, symulowanie ataków przez red team

2) Mobilne:

Sposoby ochrony:

- Szyfrowanie danych
- Korzystanie z bezpiecznego przechowywania (Keychain w iOS, Keystore w Androidzie).
- TLS – dla HTTPS
- Obfuskacja kodu
- CORS
- Patche i update’y.
- Tokenizacja sesji (np. OAuth 2.0, JWT z odpowiednim TTL).

Podatności:

Z top 10 (2024): Improper Credential Usage, Inadequate Supply Chain Security, Insecure Authentication/Authorization,Insufficient Input/Output Validation, Insecure Communication, Inadequate Privacy Controls, Insufficient Binary Protections, Security Misconfiguration, Insecure Data Storage, Insufficient Cryptography, zahardkodowane klucze i hasła, brak szyfrowania danych,

Ataki:

Wykorzystywanie podatnych wersji frameworków i SDK, MITM, inżynieria wsteczna, phishing, Overlay attack – nakładanie fałszywego interfejsu na oryginalną aplikację – np. po zdobycie uprawnień, Skimming – nieautoryzowanym uzyskanie danych z kart płatniczych, root/jailbreak (IOS) – procesy odblokowujące ograniczenia narzucone przez producentów urządzeń. (np. nie trzeba z App Store'a pobierać)

Ocena bezpieczeństwa:

- SAST/DAST
- testy penetracyjne
- testowanie API – czy dane są bezpiecznie przesyłane między aplikacją i serwerem

Bezpieczne przechowywanie i przetwarzanie danych (przechowywanie haseł, walidacja danych, dobre i złe praktyki programistyczne itp.)

1) Sposoby przechowywania haseł (więcej patrz powiązane pytania):

- Haszowanie/szyfrowanie – lepsze jest haszowanie
- Solenie, pieprzenie haseł przed zahaszowaniem (sól na koniec, pieprz na początek)

2) Best practices dla haseł i logowania:

- Odpowiednie reguły haseł (np. wymagana długość)
- Ograniczenie prób logowania
- Okres ważności haseł (w banku nie ma potrzeby, bo jak ktoś się wjebie to od razu cały hajs wypłaci, a nie zależy mu na tym, żeby mieć dostęp przez długi czas)
- Automatyczne wylogowanie podczas dłuższej nieaktywności
- Ograniczenia dostępu, np. po IP
- Analiza zachowań użytkowników (wykrywanie anomalii)

3) Walidacja danych

- Zapewnienie jakości danych wejściowych, można użyć gotowych mechanizmów we frameworku

Metody walidacji:

- whitelisty (zezwala na wybrane wzorce)
- blacklisty (zabrania wybranych wzorców)
- sanityzacja (wycinamy lub zamieniamy niechciane dane, np. apostrof w nazwiskach w bazie SQL)

4) Dobre praktyki programistyczne:

- **Security through obscurity (bezpieczeństwo przez niejawność)**:
    - Służy jedynie do odfiltrowania tych słabych hackerów.
        - Nie zapewnia 100% bezpieczeństwa!
    - przykład praktyk stosowanych w bezpieczeństwie teleinformatycznym, którego istotą jest ukrywanie detali dotyczących implementacji, formatów i protokołów przed potencjalnymi adwersarzami.
    - Osoby stosujące tę technikę zakładają, że nawet jeśli system posiada luki, nieznajomość błędów utrudnia przeprowadzenie ataku.
    - W środowiskach kryptograficznych i bezpieczeństwa security through obscurity jest powszechnie uważane za podejście błędne, które obniża, zamiast podwyższać, bezpieczeństwo systemu.
- Używanie własnych algorytmów (ale nie kryptograficznych!)
- Używanie niestandardowych portów
- Zaciemnianie kodu (obfuskacja)
- Wdrażaj mechanizmy zabezpieczające opisane w poprzednim pytaniu.
- Zmniejszenie obszaru, który może podlegać atakowi (zamiast inputa, hierarchiczne wyszukiwanie produktów)
- Tylko niezbędne uprawnienia (role, prawa)

- Odpowiednia obsługa błędów i wyjątków (nie podawać użytkownikowi dokładnej wiadomości o błędzie, co najwyżej kod)
- Nie ufać zewnętrznym serwisom

5) Złe praktyki programistyczne:

- Komentarze w kodzie
- Zostawienie testowych/domyślnych credentiali na prodzie
- Przeciwieństwa dobrych praktyk

Ochrona danych osobowych i technologie wzmacniające prywatność

Prywatność użytkowników systemów teleinformatycznych i techniki śledzenia

Prawo do prywatności pojawia się w:

- Powszechna Deklaracja Praw Człowieka
- Międzynarodowy Pakt Praw Obywatelskich i Politycznych
- Konwencja o Ochronie Praw Człowieka i Podstawowych Wolności
- Karta Praw Podstawowych UE
- Konstytucja RP z 1997 r.

Zagrożenia prawa do prywatności:

- Działania służb i administracji państwowej
- Zbieranie danych przez firmy
- Nadzór nad pracownikami
- Internet i telefonia komórkowa
- Monitoring wizyjny
- Rozwój biometrii

Podstawowe techniki śledzenia w systemach teleinformatycznych:

1. Pliki cookies (ciasteczka)

Pliki tekstowe zapisywane w przeglądarce użytkownika, zawierające informacje związane z odwiedzaną stroną internetową.

Rodzaje cookies:

- Sesyjne (session cookies) – tymczasowe, istniejące tylko w trakcie aktywnej sesji przeglądarki.
- Stałe (persistent cookies) – przechowywane przez dłuższy czas w urządzeniu użytkownika.
- Third–party cookies – pochodzące od podmiotów zewnętrznych (np. sieci reklamowych), pozwalające na śledzenie użytkownika między różnymi stronami.

Zastosowanie cookies:

- Utrzymanie sesji (np. koszyk w sklepie internetowym).

- Zapamiętywanie preferencji (np. ustawienia językowe).
- Śledzenie aktywności (remarketing, personalizacja reklam).

2. Tracking pixele (piksele śledzące)

- Mały, często niewidoczny obrazek (1×1 piksel) umieszczany na stronie internetowej lub w e–mailu.
- Służy do monitorowania aktywności użytkowników (np. otwarcie wiadomości, przejście na konkretną stronę).
- Może przekazywać do serwera informacje o zachowaniu użytkownika, adresie IP czy używanej przeglądarce.

3. Fingerprinting (odcisk przeglądarki/urządzenia)

- Metoda polegająca na zbieraniu unikatowych informacji o konfiguracji urządzenia użytkownika (typ przeglądarki, system operacyjny, rozdzielczość ekranu, wtyczki, strefa czasowa itp.).
- Cel: Utworzenie „profilu” (unikalnego identyfikatora) urządzenia bez użycia cookies.
- Fingerprinting jest trudny do zablokowania, ponieważ wykorzystuje cechy techniczne, które przeglądarka ujawnia w trakcie normalnego działania.

4. Adresy IP i geolokalizacja

- Adres IP może wskazać przybliżoną lub dokładną lokalizację użytkownika (w zależności od usługodawcy i metody).
- Informacje o lokalizacji (GPS, Wi–Fi, BTS) w urządzeniach mobilnych.
- Stosowane w celach analitycznych, targetowania reklam, usług geolokalizacyjnych.

5. Analiza ruchu sieciowego (np. DPI – Deep Packet Inspection)

- Zaawansowane metody analizy pakietów danych w ruchu sieciowym.
- Możliwość wyodrębniania szczegółowych informacji z treści przesyłanych w Internecie.
- Stosowane m.in. przez niektórych dostawców usług internetowych, służby państwowe czy firmy do monitorowania zgodności z polityką bezpieczeństwa.

6. Aplikacje mobilne i ich uprawnienia

- Aplikacje na smartfony często proszą o dostęp do takich danych jak kontakty, kamera, mikrofon, lokalizacja.
- Nadużywanie uprawnień może prowadzić do śledzenia aktywności użytkowników, np. w celach marketingowych lub nawet szpiegowskich.

Jak firmy i instytucje wykorzystują gromadzone dane:

1. Marketing i reklama spersonalizowana

- Profilowanie użytkowników na podstawie ich historii przeglądania (cookies, fingerprinting).
- Wyświetlanie reklam dopasowanych do zainteresowań użytkownika (systemy DSP, SSP, Google Ads, Facebook Ads itp.).

2. Analiza zachowań użytkowników (web analytics)

- Narzędzia takie jak Google Analytics, Hotjar, Piwik/Matomo.
- Pozwalają na mierzenie ruchu, analizy demograficzne, śledzenie ścieżki użytkownika na stronie, „mapy ciepła” (heatmap).

3. Personalizacja treści

- Serwisy streamingowe (np. Netflix) i portale społecznościowe (np. Facebook) rekomendują treści na podstawie historii oglądania, polubień, interakcji ze znajomymi.
- Algorytmy uczą się preferencji użytkownika, w efekcie tworząc bańkę informacyjną (tzw. filter bubble).

4. Inne zastosowania

- Bezpieczeństwo (np. wykrywanie nadużyć i oszustw/wyłudzeń w bankowości online).
- Zarządzanie ryzykiem (np. ocena zdolności kredytowej w oparciu o nietypowe dane).
- Cele polityczne (np. mikrotargetowanie w kampaniach politycznych).

Metody ochrony prywatności

- Świadome zarządzanie plikami cookies i uprawnieniami
- Anonimizery i VPN (Virtual Private Network)
- Tor (The Onion Router)
- Przeglądarki i wtyczki nastawione na prywatność
- Szyfrowanie komunikacji
- Korzystanie z protokołu HTTPS
- Szyfrowanie end–to–end (E2EE) w komunikatorach
- Używanie poczty e–mail z szyfrowaniem PGP
- Dbałość o aktualizacje i bezpieczeństwo urządzenia
- Odpowiednie ustawienia w mediach społecznościowych

Ochrona i przetwarzanie danych osobowych oraz informacji niejawnych

Dane osobowe – Wszelkie informacje dotyczące zidentyfikowanej lub możliwej do zidentyfikowania osoby fizycznej. Tożsamość można określić bezpośrednio lub pośrednio (kombinacja informacji, cech, np. imię, nazwisko, PESEL, adres IP).

Nie mamy do czynienia z danymi osobowymi wówczas, gdy informacja dotyczy instytucji (np. firmy, organizacji), grupy osób, osoby zmarłej lub fikcyjnej

Przetwarzanie danych – operacje/zestaw operacji wykonywanych na danych osobowych

Administrator – samodzielnie lub wspólnie z innymi ustalający cele i sposoby przetwarzania danych osobowych

Podmiot przetwarzający – przetwarza dane osobowe w imieniu administratora

Materialny zakres – każdy rodzaj danych

RODO ustanawia przepisy o **ochronie osób fizycznych w związku z przetwarzaniem danych osobowych** oraz przepisy o **swobodnym**

przepływie danych osobowych. **Przetwarzanie ma ścisły związek z prywatnością**.

Korzyści RODO dla przedsiębiorstw:

- Zaufanie pracowników,
- Ewentualne obniżenie kosztów przechowywania starych danych

Korzyści dla ludzi:

- Większa kontrola nad danymi,
- Większe bezpieczeństwo,
- Większa pewność prawna i efektywność (stosowanie + sankcje)

W przepisach RODO sformułowanych zostało 7 zasad przetwarzania danych osobowych:

1) **Zgodności z prawem, rzetelności i przejrzystości**
    - Świadome wyrażenie zgody na wszystkie cele przetwarzania
    - Zgoda może być tak samo łatwo wycofana
    - **Przejrzystość = informowanie**
2) **Ograniczenia celu przetwarzania danych**
    - Dane mogą być zbierane wyłącznie w konkretnych, wyraźnych i prawnie uzasadnionych celach
3) **Minimalizacji danych**
    - Nie przetwarzać więcej informacji o osobie, niż jest to niezbędne.
4) **Prawidłowości danych**
    - Dane muszą być prawidłowe i w razie potrzeby uaktualniane;
5) **Ograniczenia przechowania danych**
    - Przechowywane w formie umożliwiającej identyfikację osoby, której dane dotyczą, przez okres nie dłuższy, niż jest to niezbędne do celów, w których dane te są przetwarzane;
6) **Integralności i poufności danych**
    - Przetwarzane w sposób zapewniający odpowiednie bezpieczeństwo danych osobowych,
    - Ochrona przed niedozwolonym lub niezgodnym z prawem dostępem, przetwarzaniem
7) **Rozliczalności**
    - Administrator jest odpowiedzialny za przestrzeganie przepisów i musi być w stanie wykazać ich przestrzeganie

Ustawa o ochronie danych osobowych (w Polsce) – uzupełnia RODO i reguluje m.in. organizację krajowego organu nadzorczego (**UODO – Urząd Ochrony Danych Osobowych**).

Zasady przetwarzania danych:

- **Zgoda osoby**, której dane dotyczą.
- **Przetwarzanie danych zgodnie z określonym celem**.
- **Minimalizacja danych** – zbieranie tylko tych danych, które są niezbędne.

Prawa osób, których dane dotyczą:

- Prawo **dostępu do danych** (informacja, jakie dane są przetwarzane).

- Prawo do **sprostowania danych** (poprawianie nieprawidłowych/nieaktualnych informacji).
- Prawo do **usunięcia danych** („prawo do bycia zapomnianym”).
- Prawo do **ograniczenia przetwarzania**.
- Prawo do **przenoszenia danych** (w formacie elektronicznym).
- Prawo do **sprzeciwu wobec przetwarzania** (m.in. na potrzeby marketingu bezpośredniego).

Bezpieczeństwo przetwarzania danych osobowych:

1) **Środki techniczne i organizacyjne**
   - Szyfrowanie (np. danych w bazie, transmisji – protokół HTTPS/TLS).
   - Kontrola dostępu (systemy uwierzytelniania, autoryzacji, rejestracja logów).
   - Segmentacja sieci i ograniczanie obszarów dostępu.
   - Regularne kopie zapasowe (backup) i testy przywracania.
   - Polityki bezpieczeństwa informacji, szkolenia pracowników, procedury zarządzania incydentami.
2) **Ocena ryzyka i DPIA (Data Protection Impact Assessment)**
   - Wymóg przeprowadzenia oceny skutków przetwarzania dla ochrony danych, zwłaszcza w przypadku nowych technologii, systemów lub dużej skali przetwarzania.
   - Identyfikacja i minimalizacja potencjalnych zagrożeń (np. wycieków, nieuprawnionego dostępu).
3) **Rejestrowanie czynności przetwarzania**
   - Obowiązek prowadzenia rejestrów (kto, gdzie, jakie dane, w jakim celu przetwarza).
   - Pomaga w spełnieniu zasady rozliczalności.
4) **Incydenty naruszenia ochrony danych**
   - Obowiązek zgłoszenia incydentu do organu nadzorczego (UODO) w ciągu 72 godzin od stwierdzenia.
   - W niektórych przypadkach także powiadomienie osób, których dane dotyczą (jeśli istnieje wysokie ryzyko naruszenia praw lub wolności tych osób).

Informacje niejawne – informacje, których nieuprawnione ujawnienie może spowodować szkodę dla państwa lub określonej instytucji.

Informacjami niejawnymi są np. dane dotyczące obronności, bezpieczeństwa państwa, spraw zagranicznych czy też organizacji i sposobu działania organów władzy publicznej (jeżeli uznano je za niejawne).

Środki ochrony:

1) **Kontrola dostępu**:
   - Role i uprawnienia ograniczające dostęp do danych.
2) **Szyfrowanie danych**:
   - Algorytmy szyfrujące (np. AES–256) stosowane w transmisji i przechowywaniu danych.
3) **Bezpieczne przechowywanie**:
   - Fizyczne zabezpieczenia (np. sejfy).
   - Logiczne zabezpieczenia (np. systemy DLP – Data Loss Prevention)

1) Rozwiązanie z dziedziny zabezpieczeń, które identyfikuje przypadki niebezpiecznego lub niewłaściwego udostępniania, przenoszenia lub używania poufnych danych i pomaga zapobiec ich występowaniu

Klauzule tajności (w Polsce)

- „**ściśle tajne**” – najwyższa klauzula, której ujawnienie może wyrządzić wyjątkowo poważną szkodę dla państwa.
- „**tajne**” – ujawnienie może spowodować poważną szkodę dla państwa.
- „**poufne**” – ujawnienie może spowodować szkodę dla państwa.
- „**zastrzeżone**” – ujawnienie może być szkodliwe dla interesu państwa, ale na niższym poziomie ryzyka.

Wycieki danych

Wyciekiem danych nazywamy każdą sytuację, w której informacja poufna lub chroniona (np. dane osobowe, finansowe, handlowe) trafia w niepowołane ręce, bądź staje się publicznie dostępna bez zgody i wiedzy jej właściciela. Może do niego dojść na skutek:

- Działania celowego (atak hakerski, sabotaż wewnętrzny).
- Błędu lub zaniedbania (np. błędna konfiguracja bazy danych, przypadkowe wysłanie pliku do niewłaściwego odbiorcy).
- Luki w oprogramowaniu (brak aktualizacji, niezałatane podatności)

Konsekwencje wycieków danych

- Straty finansowe,
- Konieczność wypłaty odszkodowań, pokrycia kosztów powiadomień osób poszkodowanych, wdrożenia środków naprawczych,
- Możliwe kary administracyjne nakładane przez organy ochrony danych osobowych (np. UODO w Polsce, zgodnie z RODO),
- Utrata reputacji i zaufania,
- Spadek wiarygodności firmy lub instytucji w oczach klientów, partnerów biznesowych,
- Trudności z pozyskaniem nowych klientów i kontrahentów,
- Konsekwencje prawne i administracyjne,
- Naruszenie przepisów o ochronie danych osobowych (np. RODO), co może skutkować wysokimi karami finansowymi (nawet do 20 mln EUR lub 4% globalnego obrotu firmy),
- Pozwy zbiorowe ze strony osób, których dane zostały ujawnione,
- Wzrost ryzyka kradzieży tożsamości.

Zapobieganie i reagowanie na wycieki danych:

Wdrażanie odpowiednich zabezpieczeń technicznych

- Szyfrowanie danych w spoczynku (na dyskach) i w trakcie przesyłania (TLS/HTTPS).
- Regularne aktualizacje oprogramowania i systemów.
- Zapory sieciowe (firewall), systemy wykrywania intruzów (IDS/IPS) i narzędzia antywirusowe.

- Segmentacja sieci, ograniczanie dostępu do baz danych tylko dla niezbędnych usług i osób.

Silne procedury organizacyjne i polityka bezpieczeństwa

- Polityka haseł (regularna zmiana, odpowiednia złożoność), uwierzytelnianie wieloskładnikowe (MFA).
- Jasno sprecyzowane zasady dostępu do danych (różne poziomy uprawnień, zasada najmniejszych przywilejów).
- Szkolenia pracowników w zakresie rozpoznawania phishingu, bezpiecznego korzystania z poczty e–mail i Internetu.

Monitoring i audyt

- Regularne testy penetracyjne (pentesty) i audyty bezpieczeństwa.
- Logowanie i analiza zdarzeń (SIEM – Security Information and Event Management).
- Bieżący monitoring anomalii w ruchu sieciowym (np. masowy transfer danych w nietypowym czasie).

Plan reagowania na incydenty (Incident Response Plan)

- Określenie kroków postępowania w przypadku wykrycia wycieku (kogo poinformować, w jaki sposób, jakie działania podjąć w pierwszej kolejności).
- Identyfikacja źródła wycieku, izolacja zainfekowanych systemów, przywracanie danych z backupów.
- Szybkie zgłoszenie do odpowiednich organów (np. UODO w przypadku naruszenia ochrony danych osobowych) i powiadomienie poszkodowanych.

Polityka backupu

- Regularne tworzenie kopii zapasowych, przechowywanie w bezpiecznej lokalizacji i testy procedur przywracania.
- W przypadku ataków ransomware czy innych form sabotażu backup umożliwia minimalizację strat.

Oprogramowanie typu **Data Loss Prevention** zabezpiecza dane, które mogą wyciec na zewnątrz na wiele różnych sposobów. Najczęstsze kanały wycieku danych to:

- pendrive, dysk przenośny (urządzenia podłączane do komputera poprzez USB),
- poczta elektroniczna,
- komunikator internetowy (opcja przesyłania plików),
- płyta CD/DVD.

Przykłady wycieków:

Marriott (2018)

- Wyciekły dane około 500 milionów gości sieci hoteli (imię, nazwisko, adres e–mail, numer telefonu, numer paszportu, dane kart kredytowych).
- Przyczyną było przejęcie przez Marriott innej sieci hotelowej (Starwood) wraz z infrastrukturą IT, w której istniała już wcześniej niezidentyfikowana luka bezpieczeństwa (infekcja trwała od 2014 roku).

PlayStation Network (2011)

- Wyciekły dane ok. 77 milionów użytkowników, w tym nazwy kont, adresy e–mail, daty urodzenia, a także potencjalnie dane kart płatniczych.
- Przyczyną były niezabezpieczone podatności w infrastrukturze sieciowej firmy Sony.
    - Atak hakerski ukierunkowany na uzyskanie dostępu do serwerów (wykorzystanie exploitów w usługach sieciowych).

Bezpieczeństwo w sieciach rozległych

Mechanizmy bezpieczeństwa w sieciach rozległych i zabezpieczenie dostępu do urządzeń i usług

Mechanizmy bezpieczeństwa w sieciach rozległych:

1) **Tunelowanie** – sposób przesyłania pakietów tworząc logiczne kanały komunikacji, działa na zasadzie enkapsulacji (pakiet w pakiecie). Przydatne, jeśli jakaś sieć pomiędzy nie wspiera konkretnego protokołu np. IPv6. Może zapewnić również szyfrowanie pakietów.
    - Przykładem tunelowania jest VPN (Virtual Private Network).
    - **IPsec** – zbiór protokołów służących implementacji bezpiecznych połączeń oraz wymiany kluczy szyfrowania pomiędzy komputerami.
    - **GRE (Generic Routing Encapsulation)** – umożliwia tunelowanie, ale sam nie zapewnia szyfrowania i uwierzytelniania.
2) **NAT – Network Address Translation** – protokół, który powstał ze względu na limitowaną ilość adresów IPv4 – zostały wydzielone sieci publiczne oraz prywatne (10.x.x.x /8, 172.16.x.x/12, 192.168.x.x/24). Transponuje adres IP sieci prywatnych na adres IP sieci publicznych (choć może również public–to–public, private–to–private). Zwiększa również bezpieczeństwo, ponieważ prawdziwy adres IP specyficznego hosta jest nieznany z zewnątrz.
    - Są trzy typu: **Static** NAT (1:1), **Dynamic** NAT (ma zakres adresów do transponowania), **PAT** (Port Address Translation – transponuje ruch wychodzący rozróżniając urządzenia na podstawie numerów portów na pojedynczy adres globalny)
3) **ACL – Access Control List** – lista reguł do kontrolowania ruchu przychodzącego i wychodzącego. Może być zastosowana na routerach i przełącznikach. Reguły są ustalane na podstawie adresów IP (standardowe), numerów portów, protokołów (rozszerzone).
4) **VLAN – Virtual Local Area Network** – logiczne segmentowanie sieci. Służy do ograniczania domeny rozgłoszeniowej. Są definiowane na przełącznikach i identyfikowane za pomocą VLAN tagów.
    - **Ważne pojęcia w kontekście bezpieczeństwa**: **native vlan**, **trunk ports, disable unused ports**, **allow proper inter–vlan communication**, **root bridge**, **VLAN ACL**, **Private VLAN**
5) **Użycie poprawnej konfiguracji protokołów rutingu:**
    - **BGP (Border Gateway Protocol)** – jest to zewnętrzny protokół rutingu pomiędzy systemami autonomicznymi (sieć lub grupa sieci z

wspólną polityką routowania) – zapewnia łączność na poziomie globalnym.
    - **OSPF (Open Shortest Path First)** – protokół rutingu w ramach jednej sieci autonomicznej, działa wewnątrz, zapewnia wydajność i niezawodność w dużych sieciach, wersja v2 wspiera mechanizmy uwierzytelniania (przesyłanie hasła w każdym pakiecie, ale plain text, podpisywanie MD5 lub SHA–256 i shared key).
    - **RIP – Routing Information Protocol** – używany podobnie jak OSPF, ale zlicza maksymalnie 15 przeskoków, stosowany w małych sieciach
6) **Sieci MPLS/BGP VPN** – umożliwiają w skalowalny sposób łączyć rozproszone geograficznie lokalizacje – tworzy wirtualne prywatne sieci rozległe na podstawie infrastruktury MPLS.
    - MPLS (Multiprotocol Label Switching) – trasowanie pakietów zostało zastąpione przez przełączanie etykiet, umożliwia rezerwację pasma dla przepływu ruchu, gwarantuje rozróżnienie wymagań Quality of Service i implementowanie Virtual Private Network.
    - Umożliwia zarządzanie wieloma sieciami w obrębie jednej infrastruktury MPLS.
7) **Serwery proxy** – serwer służący jako pośrednik między użytkownikiem a siecią. Działa na poziomie aplikacji, przechwytuje requesty klientów i przekazuje do odpowiednich serwerów, a następnie zwraca odpowiedzi do klientów.
    - Pozwala to na kontrolowanie polityki dostępu, filtracje treści, maskowanie, szyfrowanie i działanie jako zapora chroniąca przed atakami.
8) **Strefy DMZ** – segmenty sieci, które działają jako bufor między siecią wewnętrzną a zewnętrzną. Zapewniają izolację zasobów np. serwerów WWW, czy poczty. Ograniczają dostęp ze względu na protokoły czy porty, umożliwiają wykrywanie intruzów, zabezpieczają przed atakami (pivoting)

Zabezpieczenie dostępu do urządzeń i usług:

1) **AAA – Authentication, Authorization, Accounting** – zestaw mechanizmów bezpieczeństwa do zarządzania dostępem i kontrolą użytkowników. **Authentication** – sprawdzanie tożsamości, **Authorization** – przyznawanie odpowiednich uprawnień i ról, **Accounting** – monitorowanie i rejestrowanie działań. W ramach działania mogą korzystać z lokalnej bazy użytkowników lub specjalnych serwerów logowania np. RADIUS
2) **Korzystanie z bezpiecznych protokołów**, np. SSH zamiast telnetu albo protokołu SNMPv3
3) **Monitorowanie oraz logowanie,** np. za pomocą sysloga
4) **Inne (w cisco)**:
    - encrypted password,
    - ustawienie secret password,
    - proper console vty access with passwords,
    - clear text password encryption,
    - enhanced virtual login,
    - exec–timeout (rozłącza sesje po czasie nieaktywności),
    - włączenie logów,

- odpowiednie konta administratorów z minimalizacją potrzebnych uprawnień (privileged mode),
- login banner,
- zmniejszenie maksymalnej ilości ponownych prób logowania,
- secure Cisco IOS image,
- Cisco AutoSecure

Bezpieczeństwo poprzez zapory sieciowe

- **Klasyczny firewall**. Zapewnia filtrację pakietów na podstawie adresów IP, numerów portów, protokołów. Wyróżnia się zapory stanowe i bezstanowe.
- **WAF (Web Application Firewall)** – zapory aplikacyjne chroniące aplikacje internetowe np. przed atakami typu SQL injection czy XSS
- **NGFW – Next Generation Firewall** – firewall nowej generacji, łączą funkcję klasycznego firewalla z zaawansowanymi funkcjami np. Deep Packet Inspection lub systemów IPS, integracja z systemami anti malware
- **zapory chmurowe** – Zapory chmurowe to rozwiązania zabezpieczające, które działają w środowisku chmurowym i oferują ochronę przed zagrożeniami, zapewniając kontrolę nad ruchem przychodzącym i wychodzącym w sieci wirtualnej.
- **zone–based policy Firewall** – jest przykładem firewalla stanowego, używa podejścia opartego na strefach do kontrolowania ruchu. Sieć jest dzielona na strefy (np. wewnętrzne, zewnętrzna DMZ), Każda strefa ma określone swoje reguły i polityki zabezpieczeń.
- **ACL** – Listy Kontroli Dostępu. Standardowe, rozszerzone
- **VACL – VLAN ACL** – podobne do ACL, ale stosowane na przełącznikach warstwy 2 w celu kontrolowania ruchu w obrębie określonego VLAN–u
- **IDS/IPS** – systemy wykrywania i zapobiegania włamaniom, które monitorują ruch sieciowy i analizują go w poszukiwaniu nieautoryzowanych działań. IDS wykrywa zagrożenia i generuje alerty, podczas gdy IPS aktywnie blokuje podejrzany ruch.

Protokoły bezpieczeństwa i sieci VPN

Protokoły bezpieczeństwa:

1) **SSL/TLS** – Protokoły zabezpieczające, które służą do szyfrowania danych przesyłanych między klientami a serwerami. Używane głównie w połączeniach HTTPS, zapewniają poufność, integralność i uwierzytelnianie
2) **IPSec** – zbiór protokołów służących implementacji bezpiecznych połączeń oraz wymiany kluczy szyfrowania pomiędzy komputerami.
    - Składa się z dwóch kanałów (wymiany kluczy oraz przesyłu danych).
    - Protokoły IPSec działają na zasadzie enkapsulowania, gdzie oryginalny pakiet jest szyfrowany i otrzymuje nowy nagłówek protokołu IPsec i w takiej formie jest przesyłany przez sieć.
    - IPsec charakteryzuje się **jednokierunkowością**, każdy kanał ma niezależne parametry (klucze, licznik sekwencji, SPI (Security Parameters index).
    - IPSec wykorzystuje **ESP (Encapsulating Security Payload)** do zapewnienia szyfrowania i uwierzytelniania danych oraz **AH (Authentication Header)**, który zapewnia tylko uwierzytelnianie.

- Tryby IPSec:
    1) **Tryb Transportowy (Transport Mode)**:
        - Szyfrowanie tylko zawartości pakietu, nagłówek IP pozostaje niezmieniony.
    2) **Tryb Tunelowy (Tunnel Mode)**:
        - Szyfrowanie całego pakietu (nagłówek i dane), a nowy nagłówek IP jest dodawany.

3) **IKE – Internet Key Exchange** – protokół używany w ramach IPSec do negocjowania i wymiany kluczy szyfrujących oraz ustalania parametrów sesji

Sieci VPN

Sieci VPN umożliwiają tworzenie bezpiecznych, zaszyfrowanych połączeń przez publiczne sieci, jak Internet, aby umożliwić zdalny dostęp do zasobów sieciowych lub połączenie ze zdalnymi lokalizacjami. Zapewniają:

- szyfrowanie danych,
- uwierzytelnianie (hasła, certyfikaty),
- anonimowość i prywatność – maskują prawdziwy adres IP,
- zdalny dostęp – np. do zasobów firmowych z dowolnego miejsca na świecie.

Typy VPN:

1) **Site–to–Site** – tworzy bezpieczne połączenie między dwoma sieciami LAN
2) **PPTP – Point–to–point** – tworzy tunel i przechwytuje wysyłane dane. Ten rodzaj VPN wykorzystywany jest przede wszystkim w pracy zdalnej.
3) **L2TP – Layer to Tuneling Protocol** – tworzy tunel pomiędzy dwoma punktami połączeń L2TP VPN, a inny protokół taki, jak IPsec szyfruje dane i koncentruje się na zapewnieniu komunikacji między tunelami.

Techniki VPN:

1) **Sieci MPLS/BGP VPN** – Jest to metoda tworzenia sieci VPN, która wykorzystuje MPLS do zapewnienia wysokiej wydajności i rutingu między różnymi lokalizacjami. BGP jest używany do wymiany informacji o trasach między routerami w sieci.
2) **GETVPN – Group Encrypted Transport VPN** – stosowane przez Cisco, umożliwia szyfrowanie ruchu między urządzeniami w jednej grupie (np. oddziałami firm) bez potrzeby tworzenia indywidualnych tuneli dla każdego połączenia, wykorzystuje IPsec
3) **FlexVPN** – stosowane przez Cisco, łączy funkcje różnych protokołów VPN (np. IPSec, IKEv2, L2TP) i umożliwia łatwe skalowanie oraz konfigurowanie połączeń VPN
4) **DMVPN – Dynamic Multipoint VPN** – stosowana przez Cisco, łączy funkcje różnych protokołów VPN (np. IPSec, IKEv2, L2TP) i umożliwia łatwe skalowanie oraz konfigurowanie połączeń VPN

Bezpieczeństwo systemów komórkowych i sieci 5G

Zjawiska w kanale radiowym charakterystyczne dla radiokomunikacji ruchomej, metody wielodostępu i problematyka ochrony przed PEM

Zjawiska w kanale radiowym charakterystyczne dla radiokomunikacji ruchomej:

- **Wielodrogowość (zaniki Rayleigha/Szybkie)** – sygnał z nadajnika do odbiornika może dotrzeć więcej niż jedną drogą (np. poprzez odbicie od obiektu). Przy innej drodze, sygnał dociera do odbiornika w innym czasie co powoduje zakłócenia (sygnał się do siebie dodaje)
- **Efekt cienia radiowego (zaniki powolne)** – gdy pomiędzy nadajnikiem a odbiornikiem fal radiowych pojawia się przeszkoda (np. budynek, drzewo), wówczas fala elektromagnetyczna ulega odbiciom i ugięciom. W efekcie sygnał dociera do odbiornika po różnych drogach i z różnym opóźnieniem.
    - Z uwagi na interferencję poszczególnych ścieżek sygnału, w przestrzeni pojawiają się miejsca, gdzie sygnał ten zanika i jest nie do odebrania.
- **Przesunięcie Dopplera** – ruch nadajnika lub odbiornika powoduje zmianę częstotliwości odbieranego sygnału, co wpływa na jakość transmisji (wynika z efektu Dopplera).

Metody Wielodostępu:

- **SDMA (Space Division Multiple Access)**:
    - Tu chodzi, że w zależności, gdzie się znajdujesz to się łączysz do innej stacji bazowej (taka metoda bardziej do pominięcia)
    - **Zasób współdzielony** -> lokalizacja
- **FDMA (Frequency Division Multiple Access)**:
    - Pasmo częstotliwości jest dzielone na osobne kanały, z których każdy jest przypisany do jednego użytkownika.
    - **Zasób współdzielony** -> częstotliwość
- **TDMA (Time Division Multiple Access)**:
    - Wszyscy użytkownicy korzystają z tego samego kanału częstotliwościowego, ale w różnych przedziałach czasowych (tzw. szczelinach czasowych).
        - Zasób współdzielony -> czas
- **CDMA (Code Division Multiple Access)**:
    - Użytkownicy korzystają z tego samego pasma i transmitują jednocześnie, ale ich sygnały są kodowane unikalnymi sekwencjami pseudolosowymi.
    - **Zasób współdzielony** -> kody, a fizycznie poziom łącznego szumu/ hałas w kanale
    - Sygnał jest rozproszony na cały kanał.
        - rozpraszanie sygnału etapy:
            - I channelization (ortogonalizacja kodu),

        - II scrambling (mieszanie kodu) - dobra losowość, funkcja korelacji własnej
- **OFDMA (Orthogonal Frequency Division Multiple Access)**:
    - Pasmo jest dzielone na wiele nośnych o ortogonalnych częstotliwościach, a każda nośna jest przypisana konkretnemu użytkownikowi.
    - **Zasób współdzielony** -> częstotliwość

W skrócie:

Generacja – Nazwy – Metoda

1) **1G** – AMPS, NMT – FDMA (Częstotliwość)
2) **2G** - GSM – TDMA (Czas), FDMA (Częstotliwość) i Narrowband CDMA (kody)
3) **3G** - UMTS – Wideband CDMA (Kody) i Narrowband CDMA (Kody)
4) **4G** - LTE – Downlink: OFDMA (Częstotliwość) i Uplink: SC-FDMA (Częstotliwość)
5) **5G** - NR – OFDMA (Częstotliwość)

Problematyka ochrony przed PEM:

- Obecnie uważa się, że żadne środowisko nie uległo tak ogromnej dewastacji, jak naturalne środowisko elektromagnetyczne
- Mamy dyrektywy unijne oraz lokalne normy (Rozporządzenie Ministra Zdrowia), które określają dopuszczalne wartości natężenia PEM
- Większe zagrożenie mogą stanowić stacje ruchome niż bazowe
- Paradoksalnie lepiej mieć w pobliżu stację bazową swojego operatora, bo im gęstsza struktura sieci tym mniejsze wartości PEM emitowanych zarówno ze stacji bazowych, jak i terminali abonenckich

Czynniki wpływające na realną przepływność transmisji danych dostępnej dla użytkownika sieci komórkowej i funkcja układów wieloantenowych

Czynniki wpływające na realną przepływność transmisji danych w sieci komórkowej

1. Technologia sieci komórkowej

- **Generacja sieci:** Prędkość transmisji rośnie wraz z postępem technologii (np. 3G, 4G LTE, 5G).
- **Szerokość pasma:** Szersze pasmo pozwala na przesyłanie większej ilości danych jednocześnie.
- **Modulacja i kodowanie:** Zaawansowane techniki modulacji (np. 64-QAM, 256-QAM) umożliwiają wyższą przepływność.

2. Warunki radiowe

- **Moc sygnału:** Im bliżej użytkownik znajduje się stacji bazowej, tym lepsza jakość sygnału.
- **Zakłócenia:** Inne urządzenia, przeszkody terenowe (np. budynki, drzewa) oraz warunki atmosferyczne mogą osłabiać sygnał.

- **Współdzielenie częstotliwości:** Użytkownicy korzystający z tych samych częstotliwości mogą wzajemnie zakłócać swoje transmisje.

3. Zarządzanie ruchem w sieci

- **Obciążenie sieci:** W godzinach szczytu, gdy wielu użytkowników korzysta z tej samej stacji bazowej, przepływność dla pojedynczego użytkownika może się zmniejszyć.
- **Jakość usług (QoS):** Mechanizmy priorytetyzacji mogą ograniczać prędkość transmisji dla mniej istotnych aplikacji.

4. Urządzenie użytkownika

- **Zdolności techniczne urządzenia:** Liczba obsługiwanych pasm, kategoria LTE/5G i liczba anten w urządzeniu wpływają na maksymalną możliwą prędkość.
- **Oprogramowanie:** Efektywność stosowanego systemu operacyjnego oraz aplikacji obsługujących transmisję danych.

Funkcja układów wieloantenowych

Układy wieloantenowe (ang. **MIMO – Multiple Input Multiple Output**) odgrywają kluczową rolę w zwiększaniu przepływności transmisji danych oraz poprawie jakości połączeń w sieciach komórkowych. Ich funkcje można podzielić na następujące:

1. Zwiększenie przepustowości

- MIMO umożliwia równoczesne przesyłanie wielu strumieni danych między urządzeniem a stacją bazową, co znacznie zwiększa efektywną przepływność.

2. Poprawa odporności na zakłócenia

- Dzięki zastosowaniu wielu anten, układy MIMO mogą wykorzystywać różnorodność przestrzenną, co zmniejsza wpływ zakłóceń i utraty sygnału spowodowanej odbiciami.

3. Efektywne wykorzystanie widma

- MIMO zwiększa efektywność wykorzystania dostępnych częstotliwości, co jest kluczowe w środowiskach o dużym zagęszczeniu użytkowników.

4. Techniki beamformingu

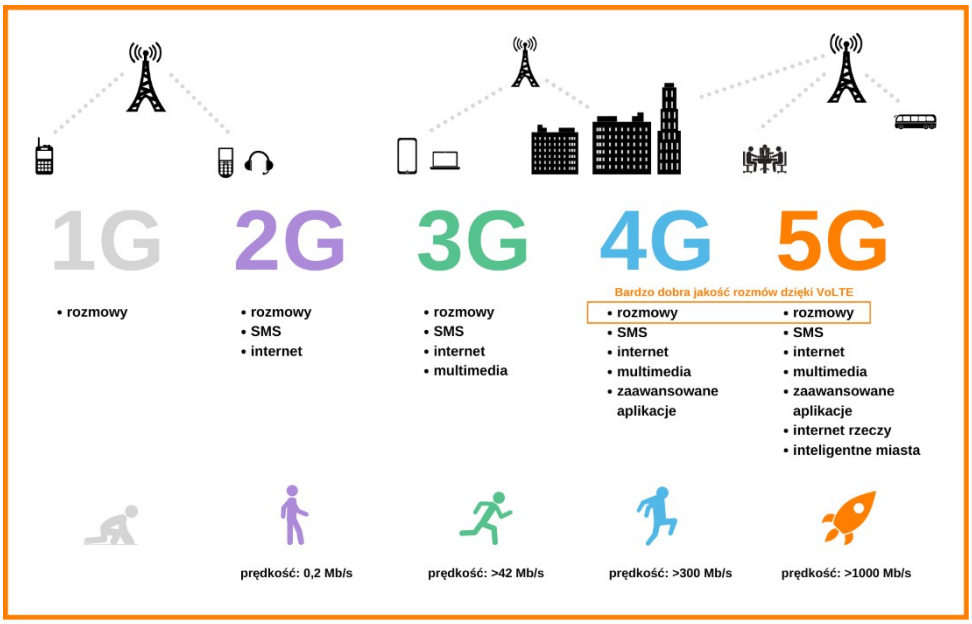
- Dzięki odpowiedniemu sterowaniu fazą i amplitudą sygnałów, MIMO może kierować sygnał w stronę konkretnego użytkownika, co zwiększa jakość transmisji i minimalizuje zakłócenia z innymi użytkownikami.

5. Adaptacja do warunków propagacyjnych

- Zaawansowane algorytmy umożliwiają dynamiczne dostosowanie liczby aktywnych strumieni danych i charakterystyk anten do zmieniających się warunków radiowych.

Ewolucja systemów 4G – 5G

Technology → / Features ↓	1G	2G	3G	4G	5G
Start/ Deployment	1980's	1990's	2000's	2010's	2020's
Data Bandwidth	2kbps	64kbps	2Mbps	1 Gbps	Higher than 1Gbps
Technology	Analog Cellular Technology	Digital Cellular Technology	Broadband with CDMA, IP technology	Unified IP & seamless combination of broadband, LAN, WAN & WLAN	Unified IP & seamless combination of broadband, LAN, WAN, WLAN & WWWW
Service	Mobile Telephony (Voice)	Digital voice, SMS, Higher capacity packetized data	Integrated high quality audio, video and data	Dynamic Information access, Wearable devices	Dynamic Information access, Wearable devices with AI Capabilities
Multiplexing	FDMA	TDMA, CDMA	CDMA	CDMA	CDMA
Switching	Circuit	Circuit, Packet	Packet	All Packet	All Packet
Core Network	PSTN	PSTN	Packet N/W	Internet	Internet



4G – Czwarta generacja sieci mobilnych

Charakterystyka:

- **Szybka transmisja danych:** 4G wprowadziło znaczący wzrost prędkości przesyłu danych, umożliwiając korzystanie z zaawansowanych usług multimedialnych i aplikacji internetowych.
- **Duża pojemność sieci:** Zwiększona liczba jednoczesnych połączeń oraz lepsza jakość usług dzięki efektywniejszemu zarządzaniu zasobami sieciowymi.
- **Wprowadzenie w Polsce:** Pierwsze komercyjne sieci 4G LTE uruchomiono w Polsce w 2010 roku.

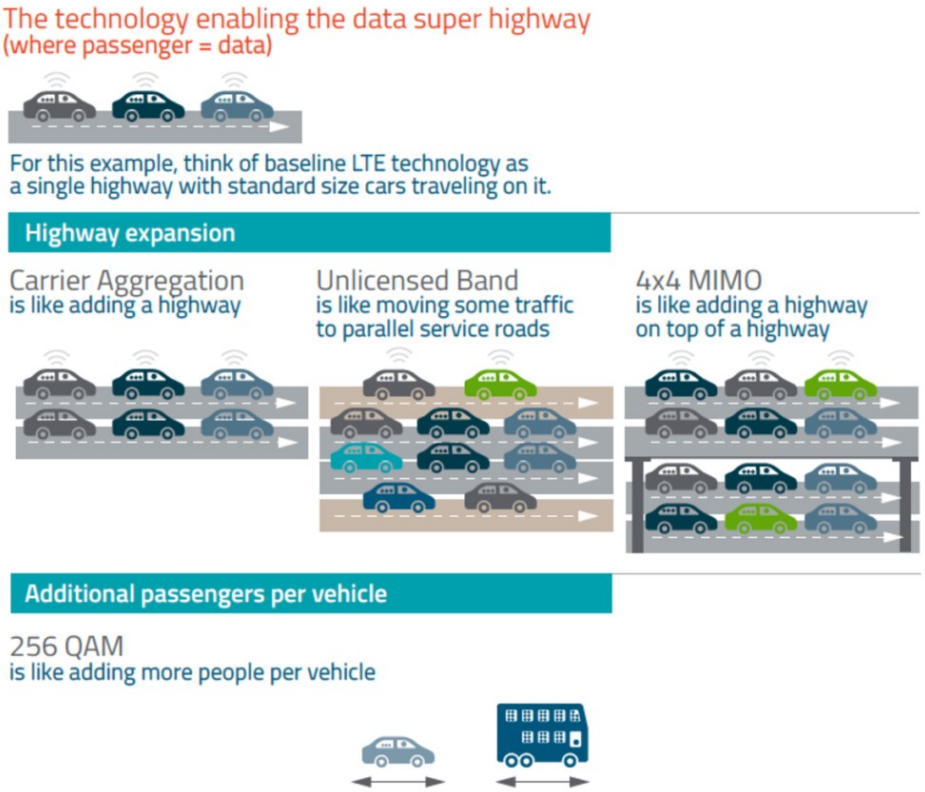
Kluczowe technologie:

- **LTE (Long Term Evolution):**
  - **Techniki transmisji:** OFDMA (Orthogonal Frequency Division Multiple Access) w downlinku oraz SC-FDMA (Single Carrier Frequency Division Multiple Access) w uplinku.
  - **SAE (System Architecture Evolution):** Uproszczona i zoptymalizowana architektura sieciowa, oparta na protokole IP, zwiększająca efektywność przesyłu danych.
- **LTE-Advanced:**
  - **Wersje:** Rozwój standardu LTE w ramach 3GPP Release 10 i kolejnych, wprowadzający m.in. agregację nośnych i zaawansowane techniki MIMO.
- **LTE-Advanced Pro:**
  - **Rozszerzenia:** Dalsza ewolucja LTE, znana również jako "4.5G", oferująca jeszcze wyższe prędkości transmisji danych oraz wsparcie dla nowych usług i urządzeń.

Komponenty sieci:

- **EPC (Evolved Packet Core):**
  - **Architektura All-IP:** Całkowite przejście na protokół IP, umożliwiające integrację różnych usług, takich jak transmisja głosu i danych, w jednej sieci.
  - **PSS (Packet-Switched Streaming):** Obsługa usług strumieniowych w trybie pakietowym, zapewniająca lepszą jakość i efektywność transmisji.
- **E-UTRAN (Evolved Universal Terrestrial Radio Access Network):**
  - **Technika OFDMA:** Wykorzystanie OFDMA w warstwie dostępu radiowego, pozwalające na efektywne zarządzanie zasobami częstotliwościowymi i zwiększenie przepustowości.
- **VoLTE (Voice over LTE):** Technologia umożliwiająca realizację połączeń głosowych w sieci LTE, zapewniająca wyższą jakość dźwięku i krótszy czas zestawiania połączeń.

Rozwój technologiczny – "4.5G":



- **Techniki:**
  - **CA (Carrier Aggregation):** Agregacja wielu nośnych w celu zwiększenia przepustowości i prędkości transmisji danych.
  - **MIMO (Multiple Input Multiple Output):** Wykorzystanie wielu anten nadawczych i odbiorczych dla poprawy jakości i szybkości transmisji.
  - **CoMP (Coordinated Multipoint):** Koordynacja transmisji z wielu punktów dostępowych w celu redukcji zakłóceń i poprawy wydajności sieci.
  - **Relays:** Stosowanie przekaźników sygnału w celu rozszerzenia zasięgu i poprawy jakości połączeń w trudno dostępnych miejscach.
- **Infrastruktura:**
  - **HetNets (Heterogeneous Networks):** Integracja różnych typów sieci (np. makrokomórki, mikrokomórki) w celu zwiększenia pojemności i zasięgu.
  - **Small Cells:** Małe komórki o niewielkim zasięgu, stosowane w miejscach o dużym zagęszczeniu użytkowników dla poprawy jakości usług.
- **Rozszerzenia LTE:**
  - **LTE-U (Unlicensed):** Wykorzystanie nielicencjonowanych pasm częstotliwości do transmisji danych, zwiększające dostępne zasoby spektrum.
  - **LAA (Licensed Assisted Access):** Połączenie licencjonowanych i nielicencjonowanych pasm w celu zwiększenia przepustowości.

    - **eLAA (enhanced LAA):** Ulepszone mechanizmy LAA dla jeszcze lepszej integracji pasm częstotliwości.

- **Nowe zastosowania:**
    - **LTE-M (LTE for Machines):** Optymalizowana wersja LTE dla urządzeń IoT o niskim poborze mocy i mniejszych wymaganiach dotyczących przepustowości.
    - **eMTC (enhanced Machine Type Communication):** Rozszerzone możliwości komunikacji maszynowej, wspierające większą liczbę urządzeń i lepszą efektywność energetyczną.
    - **NB-IoT (Narrowband IoT):** Technologia wąskopasmowa dla Internetu Rzeczy, umożliwiająca masowe podłączenie urządzeń o niskim zużyciu energii.

- **SON (Self-Organizing Networks):** Automatyzacja zarządzania siecią, obejmująca samokonfigurację, samonaprawę i samooptymalizację, co prowadzi do zwiększenia efektywności operacyjnej.

5G – Piąta generacja sieci mobilnych

Charakterystyka:

- **Optymalizacja sieci:** 5G zostało zaprojektowane w celu optymalizacji wydajności sieci w różnych obszarach zastosowań. System 5G oferuje lepszą jakość usług, szybszą transmisję danych, niższe opóźnienia oraz większą niezawodność. Jedną z kluczowych cech tej technologii jest jej elastyczność i dostosowanie do różnorodnych potrzeb użytkowników, takich jak wysoka przepustowość, niskie opóźnienia i wsparcie dla dużej liczby urządzeń.

- **Zastosowania:** 5G obejmuje różne scenariusze zastosowań, w tym:
    - **Ultra-wysoka przepustowość (eMBB - Enhanced Mobile Broadband):** Umożliwia przesyłanie dużych ilości danych z prędkościami dochodzącymi do gigabitów na sekundę, co jest istotne dla aplikacji takich jak VR (Virtual Reality), AR (Augmented Reality) oraz transmisja wideo w jakości 4K/8K.
    - **Niskie opóźnienia (URLLC - Ultra-Reliable Low-Latency Communications):** Niskie opóźnienia (<1 ms) umożliwiają zastosowanie w krytycznych aplikacjach, takich jak zdalne sterowanie pojazdami autonomicznymi czy zautomatyzowane fabryki.
    - **Masowe połączenia IoT (mMTC - massive Machine Type Communications):** Sieć 5G umożliwia masowe podłączenie urządzeń w Internecie Rzeczy (IoT), co pozwala na rozwój inteligentnych miast, monitorowanie środowiska, czy zautomatyzowane systemy rolnicze.

Kluczowe technologie:

- **New Radio (NR):** Jest to nowy interfejs radiowy zaprojektowany specjalnie dla 5G. NR łączy różne pasma częstotliwości (od niskich po bardzo wysokie

pasma milimetrowe), umożliwiając sieciom 5G oferowanie wysokiej przepustowości, dużej elastyczności oraz niskich opóźnień.

- **Massive MIMO (Massive Multiple Input, Multiple Output):** Technologia ta rozwija standardowe MIMO poprzez wykorzystanie ogromnej liczby anten w stacjach bazowych, co pozwala na lepsze wykorzystanie dostępnych zasobów częstotliwościowych i zwiększenie pojemności sieci.
- **Beamforming:** Technologie formowania wiązki, takie jak beamforming, poprawiają jakość sygnału przez ukierunkowanie transmisji na konkretne urządzenie, co zmniejsza interferencje i zwiększa efektywność energetyczną.
- **Agregacja nośnych i wykorzystanie pasm milimetrowych:** Umożliwiają sieciom 5G wykorzystanie dużych pasm częstotliwości w celu osiągnięcia dużych prędkości transmisji danych. Wykorzystanie pasm milimetrowych w zakresie 24 GHz i wyższym zapewnia ogromne możliwości przepustowości, choć wiąże się to z mniejszym zasięgiem.

Architektura sieci 5G:

- **Wirtualizacja i przetwarzanie w chmurze:** W 5G wprowadzono koncepcję wirtualizacji sieci, która umożliwia dynamiczne dostosowywanie zasobów sieciowych oraz centralizację zarządzania siecią w chmurze. Tego rodzaju podejście zwiększa elastyczność, pozwalając operatorom na szybkie uruchamianie nowych usług.
- **Oddzielenie funkcji sieciowych od sprzętu:** Architektura 5G wprowadza w pełni zdefiniowaną wirtualizację, co pozwala na separację funkcji sieciowych od sprzętu. Dzięki temu możliwe jest tworzenie sieci opartej na oprogramowaniu (SDN – Software-Defined Networking) oraz wykorzystanie chmurowych technologii przetwarzania danych (NFV – Network Function Virtualization).
- **Edge Computing:** Dzięki lokalizacji obliczeń blisko użytkownika, przetwarzanie danych odbywa się w tzw. "edge", co pozwala na jeszcze szybsze przetwarzanie informacji i mniejsze opóźnienia w dostępie do usług.

Zalety 5G:

- **Szybsze prędkości:** Sieć 5G oferuje transfery danych nawet do 20 Gb/s (w teorii), co jest wielokrotnie szybsze niż w przypadku 4G. Umożliwia to np. strumieniowanie filmów w jakości 8K czy korzystanie z zaawansowanych technologii AR/VR.
- **Niskie opóźnienia:** Zmniejszenie opóźnienia (do 1 ms) pozwala na zastosowanie 5G w krytycznych aplikacjach, jak zdalne operacje chirurgiczne, samochody autonomiczne czy przemysł 4.0.
- **Skalowalność i wsparcie dla IoT:** 5G umożliwia podłączenie ogromnej liczby urządzeń do sieci (do miliona urządzeń na km²), co jest kluczowe dla masowego rozwoju Internetu Rzeczy.

Zastosowania 5G:

- **Przemysł 4.0**
- **Pojazdy autonomiczne**
- **Inteligentne miasta**

Szpiegostwo przemysłowe

Szpiegostwo, wywiad i kontrwywiad

Są to kluczowe aspekty działalności związanej z bezpieczeństwem narodowym i międzynarodowym, mające na celu gromadzenie, analizowanie oraz ochranianie informacji istotnych dla państwa. Szpiegostwo, wywiad i kontrwywiad są ze sobą powiązane, tworząc kompleksowy system działań obronno–ofensywnych.

Szpiegostwo

- Polega na **tajnym pozyskiwaniu informacji**, które są **chronione**, z reguły **wbrew woli** ich właściciela. Jest to działalność wykonywana na rzecz **państw**, **organizacji międzynarodowych** lub **innych grup**, która koncentruje się na zdobywaniu danych o:
    - Działalności politycznej,
    - Strategiach wojskowych,
    - Planach gospodarczych,
    - Badaniach naukowych i technologicznych.
- **Celem szpiegostwa** jest uzyskanie przewagi strategicznej, neutralizacja zagrożeń lub prowadzenie działań wywiadowczych w celach prewencyjnych.

Wywiad

- Ma na celu **zdobywanie, analizowanie i przetwarzanie informacji** na potrzeby **państwa** lub **organizacji międzynarodowych**
- Kolokwialnie mówiąc "**agencja przestępcza działająca legalnie na zlecenie rządu**" ~prowadzący
- Zadania wywiadu:
    - Identyfikacja zagrożeń (np. terrorystycznych, cybernetycznych).
    - Prognozowanie działań przeciwników.
    - Ochrona interesów strategicznych.

Kontrwywiad

- Ma na celu **wykrywanie, neutralizowanie i przeciwdziałanie wrogim operacjom wywiadowczym**. Jest to **kluczowy element ochrony bezpieczeństwa wewnętrznego państwa**.
- **Zadania**:
    - **Ochrona informacji**: Zapobieganie wyciekowi danych strategicznych.
    - **Rozpracowywanie wrogich agentur**: Identyfikacja i neutralizacja szpiegów działających na terenie państwa.
    - **Zapewnienie bezpieczeństwa infrastruktury**: Ochrona kluczowej infrastruktury państwowej, takiej jak sieci energetyczne,

systemy komunikacji, instytucje wojskowe czy rządowe, przed wrogimi ingerencjami.

Działania socjotechniczne ukierunkowane na manipulację pracowników i nielegalne zdobywanie informacji

Główne techniki socjotechniczne (social engineering)

1) **Phishing**: Wysyłanie fałszywych e–maili, które wyglądają jak oficjalna korespondencja, w celu zdobycia poufnych danych, takich jak hasła, dane osobowe czy dane finansowe.
2) **Pretexting**: Wymyślanie fałszywej tożsamości lub scenariusza, np. podszywanie się pod współpracownika lub przedstawiciela instytucji, aby nakłonić ofiarę do udzielenia informacji.
3) **Baiting**: Oferowanie "nagrody" (np. darmowych usług, pendrive'ów, dostępu do informacji), która zawiera złośliwe oprogramowanie lub umożliwia zdobycie wrażliwych danych.
4) **Tailgating**: Fizyczne przedostanie się do chronionych obszarów firmy poprzez „podążanie” za pracownikiem wchodzącym na identyfikatorze.
5) **Shoulder Surfing**: Podglądanie wrażliwych informacji, np. wpisywanych haseł lub kodów PIN.
6) **Whaling** i **Spear Phishing**: Precyzyjnie ukierunkowane ataki na osoby pełniące kluczowe funkcje w firmie, np. zarząd lub kadrę kierowniczą.

Przykłady socjotechniki w praktyce:

1) Telefon od "działu IT",
2) Fałszywy e–mail z linkiem
3) Podstawiony nośnik USB

Sposoby przeciwdziałania przestępczości szpiegostwa przemysłowego

Zabezpieczenia techniczne:

1) **Systemy ochrony IT**:
   - Firewalle, systemy IDS/IPS
   - Szyfrowanie danych w spoczynku i transmisji (TLS)
2) **Kontrola dostępu**:
   - 2FA dla systemów informatycznych
   - RBAC – role based access control – ograniczenie dostępu do danych w zależności od funkcji pracownika
3) **Monitoring**:
   - Monitoring ruchu sieciowego i aktywności użytkowników
   - Regularne audyty bezpieczeństwa systemów IT.

Zabezpieczenia fizyczne:

1) **Kontrola dostępu**:
   - Karty magnetyczne, biometryka
   - Fizyczne zabezpieczenie pomieszczeń z danymi

2) **Ochrona przed podsłuchem**:
    - Regularne sprawdzanie pod kątem urządzeń podsłuchowych
    - Korzystanie z szyfrowanych komunikatorów

Zabezpieczenia organizacyjne:

1) **Szkolenia pracowników**:
    - Edukacja w zakresie rozpoznawania ataków socjotechnicznych
    - Wdrożenie zasad ochrony danych i bezpieczeństwa informacji
2) **Polityka bezpieczeństwa**:
    - Regularne aktualizacje zasad bezpieczeństwa
    - Ograniczenie dostępu do poufnych informacji na zasadzie “need to know”
3) **Weryfikacja pracowników i kontrahentów**

Testy penetracyjne

Standardy, dobre praktyki i przykłady wykonywania skutecznych testów penetracyjnych

Aspekty prawne:

1) **Zgoda na testy** – podpisana pisemna zgoda od właściciela infrastruktury, inne użyteczne umowy:
2) **NDA** – zapewniona poufność przekazywanych informacji w obydwie strony,
3) **MSA/SLA** – ogólne zasady co do pentestów, długoterminowa współpraca, warunki płatności itp.; informacje bardziej ogólne niż techniczne,
4) **SoW** – szczegółowy opis konkretnych zleceń opisanych w MSA
5) **Regulacje prawne** – testy podlegają pod **GDPR (RODO)**, **ISO 27001** (zarządzanie bezpieczeństwem informacji), **PCI DSS** (wymagania dot. bezpieczeństwa kart płatniczych), **HIPAA** (dane medyczne – dotyczy tylko USA)

Dobre praktyki:

1) Stosowanie się do wymienionych oraz innych powszechnie uznawanych standardów, umów prawnych.
2) Narzędzia – temat opisany poniżej;
3) Regularne testowanie – SSDLC (Secure Software Development Life Cycle) – okresowe testowanie, wdrażanie poprawek wzmacnia odporność na exploity, szczególnie te nowoodkryte

Kluczowe etapy:

1) **Planowanie** – co jest obiektem testów? Co można/trzeba a czego nie wolno testować?
2) **Zbieranie informacji**
    - **pasywne** – brak bezpośredniej interakcji z celem, osint, shodan, whois
    - **aktywne** – nmap, nessus, ogólna interakcja z systemem
3) **Eksploitacja** – analiza dynamiczna podatności, exploitowanie danego zakresu na maxa, wykorzystywanie narzędzi, samowolka. Ważne jest, aby

podczas eksploitacji środowisko testowe było całkowicie oddzielone od produkcyjnego, gdyż w przeciwnym razie możemy obniżyć jakość usługi lub całkiem ją sparaliżować na prodzie (tego nie chcemy).

4) **Postexploitation** i raportowanie – analiza zebranych podatności, opisanie ich w raporcie (patrz punkt 3). Tworząc dokumentację należy mieć na uwadze jej szczegółowość oraz współprace z zespołem programistów

Skanowanie w celu wyszukiwania podatności i automatyzacja testów penetracyjnych

1) **Nmap**/**Zenmap** – konsolowe/GUI narzędzie do skanowania portów, działających na nich usług oraz ich wersji.
    - Umożliwia mapowanie otwartych portów, wykrycie firewalla po drodze i inne. Posiada różne warianty skanów, np. TCP SYN/ACK, UDP, OS version.
    - Można dla efektywności załączać dodatkowe skrypty.
2) **ZAP**/**Burp**– narzędzia z wbudowanym proxy, ukierunkowane na aplikacje webowe.
    - Modyfikacja requestów pozwala na skuteczny pentest pod kątem SQL/OS injection, XSS, CSRF, braku walidacji, path traversal, Broken Access Control i inne.
    - Posiada repeater oraz intruder, które pozwalają na wysyłanie ręczne pakietu wraz z danymi zaciągniętymi z payloadu.
3) **Metasploit** – platforma do testów penetracyjnych ukierunkowana na exploitowanie systemu operacyjnego bądź starych wersji danych usług (SMB, FTP, SSH i inne).
    - Posiada pogrupowane moduły (Exploit, Payload, Post–Exploitation itp.), które automatycznie wykorzystują podatności, backdoory itp.
4) **POSTMAN** – testy API;
    - wysyłanie zapytań, tworzenie ręcznych testów, automatyzacja mocno
5) **Nikto** – wstępny skaner podatności webowych; skanuje headery i znane luki (CVE)
6) **testssl** – narzędzie analizujące certyfikat TLS/SSL strony webowej

Zabezpieczanie aplikacji i raportowanie wyników pentestów

Zabezpieczanie aplikacji:

- **Raport z pentestów** – zmitygować podatności w nim opisane,
- **Secure by Design** – myślenie o bezpieczeństwie aplikacji już na etapie projektowania równolegle z tworzeniem oprogramowania, a nie pod koniec,
- **Threat Modeling** – określanie potencjalnych wektorów ataku; ważna jest regularność wykonywania TM,
- **Zasada minimalnych uprawnień** – każdy user ma minimalne konieczne uprawnienia. Ponadto, jeśli komuś są one przydzielane na odpowiedni okres (np. dostęp do jakiejś usługi na danym IP) to należy zadbać o terminowe ich odebranie,

- **Walidacja danych wejściowych** – zapobiega atakom typu injection (SQL, OS, XSS). Część z nowych frameworków (np. Angular) już ma to w sobie, dlatego...
    - **Używanie najnowszych frameworków i regularne aktualizacje komponentów**,
- **Szyfrowanie danych** – zarówno w spoczynku (at rest) jak i w locie (in transit)
    - zapobiega podsłuchiwaniu, w przypadku nieautoryzowanego dostępu dane są wciąż bezpieczne (potencjalnie),
- **Kryptografia** – używanie silnych funkcji i algorytmów, które nie są zdeprecjonowane (AES–256, SHA–512, Argon, TLS 1.3 itp.),
- **Headery – w apkach webowych**:
    - **CSP (Content Security Policy)** – określa źródła z jakich witryn mogą być ładowane dane (media, obrazy, skrypty),
    - **HSTS (HTTP Strict Transport Security)** – wymuszenie ruchu szyfrowanego (HTTPS) przy dostępnie do strony,
    - **Flaga Secure** – zapewnia, że cookie będzie transportowane tylko szyfrowanym kanałem,
    - **Flaga htpOnly** – ciasteczko nie będzie wykorzystywane przez skrypty po stronie klienta,
- **Obfuskacja kodu** – w przypadku reverse engeenering kod pozostaje nieczytelny – nie zdradza szczegółów implementacji,
- **Szyfrowane połączenia** – HTTPS zamiast HTTP, SSH zamiast telnet, SMTPS zamiast SMTP.

Raportowanie:

Raport musi być klarowny, ustrukturyzowany, szczegółowy a jednocześnie zrozumiały dla odbiorcy (szefowie działu, programiści):

- **Wprowadzenie** – opis aplikacji, zawarte umowy, data, itd.
- **Opis podatności** – nazwa, szeroko opisana podatność (można wraz z przykładowym exploitem), rekomendacje usunięcia, priorytet (można się sugerować kalkulatorem CVSS) – wypadkowa prawdopodobieństwa oraz wspływu danego exploitu, rekomendacje techniczne
- **Podsumowanie z analizą ryzyka** – priorytetyzacja, umowa sprawdzenia łatania podatności/kolejnych testów

Programowanie sieciowe wspierające aplikacje bezpieczne

Aplikacje korzystające z protokołu TCP, UDP lub SCTP oraz problematyka ich bezpieczeństwa

1. Protokoły komunikacyjne: TCP, UDP i SCTP:

TCP (Transmission Control Protocol):

1) **Charakterystyka**:

    - Protokół połączeniowy, zapewniający niezawodność przesyłu danych.
    - Działa w trybie „punkt–punkt” (end–to–end).
    - Dane są przesyłane w sposób uporządkowany i bez utraty.
    - SOCK_STREAM
2) **Zastosowania**:
    - Aplikacje wymagające niezawodności i integralności danych, np. HTTP, FTP, SMTP.
3) **Cechy bezpieczeństwa**:
    - Mechanizm kontroli strumienia i retransmisji utraconych pakietów.
    - Problematyczne w kontekście ataków typu TCP SYN Flood.

UDP (User Datagram Protocol)

1) **Charakterystyka**:
    - Protokół bezpołączeniowy, wysyłający pakiety (datagramy) bez zapewnienia ich dostarczenia.
    - Brak retransmisji utraconych pakietów.
    - SOCK_DGRAM
2) **Zastosowania**:
    - Aplikacje czasu rzeczywistego, gdzie opóźnienia są kluczowe, np. VoIP, gry sieciowe, streaming wideo.
3) **Cechy bezpieczeństwa**:
    - Brak wbudowanych mechanizmów niezawodności i ochrony danych.
    - Podatność na ataki typu UDP Flood.

SCTP (Stream Control Transmission Protocol)

1) **Charakterystyka**:
    - Protokół hybrydowy, łączący cechy TCP i UDP.
    - Obsługuje wiele strumieni danych w jednym połączeniu.
    - SOCK_STREAM (okazjonalnie SOCK_SEQPACKET)
2) **Zastosowania**:
    - Telekomunikacja (np. sygnalizacja SS7), przesyłanie danych o wysokiej niezawodności.
3) **Cechy bezpieczeństwa**:
    - Lepsza odporność na pewne typy ataków (np. SYN Flood dzięki „cookie mechanism”).

Ogółem, TCP stary dobry protokół, nie za szybki, ale za to w miarę niezawodny, używany wszędzie. UDP dużo szybszy, bez żadnej kontroli. SCTP to w zasadzie lepsze TCP, ale się niezbyt przyjęło. Jest jeszcze QUIC, bazowany na UDP, ma potencjał być zajebisty, wprowadza go Google.

2. Problematyka bezpieczeństwa aplikacji sieciowych

Rodzaje zagrożeń:

1) **Ataki na protokoły transportowe**:
    - TCP SYN Flood: Przeciążenie serwera przez wysyłanie dużej liczby żądań SYN.
    - UDP Flood: Zalewanie aplikacji dużą ilością pakietów UDP.
2) **Podsłuchiwanie transmisji (Eavesdropping)**:

    - Ataki typu Man–in–the–Middle (MITM) na niezaszyfrowane połączenia.
3) **Fałszowanie danych**:
    - Wysyłanie zmodyfikowanych pakietów w celu wprowadzenia aplikacji w błąd.
4) **Odmowa usługi (DoS/DDoS)**:
    - Przeciążenie aplikacji lub sieci poprzez masowe zapytania.
5) **Ataki na integralność danych**:
    - Modyfikowanie danych w trakcie przesyłu.

Wyzwania dla poszczególnych protokołów:

1) **TCP**:
    - Ataki na mechanizm trójfazowego nawiązywania połączenia.
    - Podatność na przejęcie sesji (session hijacking).
2) **UDP**:
    - Brak mechanizmów weryfikacji danych.
    - Łatwość podszywania się pod nadawcę (IP spoofing).
3) **SCTP**:
    - Problematyka wstecznej kompatybilności z urządzeniami wspierającymi tylko TCP/UDP.

Mechanizmy bezpieczeństwa w programowaniu aplikacji sieciowych:

1) **SSL/TLS**:
    - Zapewnia poufność i integralność danych w aplikacjach TCP (np. HTTPS).
    - TLS over SCTP dla szyfrowania w aplikacjach SCTP (z użyciem DTLS).
2) **DTLS (Datagram Transport Layer Security)**:
    - Odpowiednik TLS dla protokołu UDP.
    - Przykład zastosowania: VPN oparty na UDP (np. OpenVPN).
3) **Rate limiting**, w szczególności dla protokołów połączeniowych.

Funkcje warstwy sieciowej i warstwy transportowej w programach sieciowych

1. Warstwa sieciowa (Network Layer):

Rola warstwy sieciowej:

- Warstwa sieciowa odpowiada za przesyłanie danych między urządzeniami w różnych sieciach. Jej główne funkcje obejmują ruting, adresowanie logiczne i fragmentację pakietów.

Funkcje warstwy sieciowej

1. Adresowanie logiczne

- Odpowiada za przypisanie unikalnych adresów logicznych urządzeniom w sieci.
- **Protokół**: IPv4, IPv6.
- **Przykład**: Adres IP 192.168.0.1 w sieci lokalnej.

2. Ruting

- Wyznacza optymalną trasę dla przesyłanych pakietów między sieciami.
- **Protokół**: OSPF, BGP, RIP.
- **Przykład**: Router wybiera trasę dla pakietu z urządzenia w Warszawie do serwera w Londynie.

3. Fragmentacja i defragmentacja pakietów

- Dzieli duże pakiety na mniejsze fragmenty, dostosowując je do rozmiarów jednostek danych w sieci.
- **Protokół**: IPv4 (fragmentacja), IPv6 (wymaga odpowiedzialności od nadawcy).

4. Przełączanie pakietów (Packet Switching)

- Obsługuje przesyłanie danych w oparciu o techniki przełączania pakietów.

5. Kontrola przepływu i błędów (w ograniczonym zakresie)

- Identyfikacja i sygnalizowanie błędów w adresach lub trasach.

Zastosowanie warstwy sieciowej w programowaniu:

W programowaniu sieciowym biblioteki, takie jak Python `socket`, umożliwiają ustawianie parametrów warstwy sieciowej, np. TTL (Time–To–Live).

Przykładowe protokoły warstwy sieciowej:

- IPv4/IPv6 – Adresowanie, ruting, fragmentacja pakietów.
- ICMP – Diagnoza problemów w sieci (np. ping, traceroute).
- OSPF, BGP – Ruting dynamiczny w sieciach rozległych.

2. Warstwa transportowa (Transport Layer)

Rola warstwy transportowej:

- Warstwa transportowa odpowiada za niezawodność przesyłu danych, podział danych na segmenty i zapewnienie właściwej komunikacji między aplikacjami. Jej głównym celem jest dostarczenie danych w sposób uporządkowany i kompletny.

Funkcje warstwy transportowej:

1. Multiplikacja i demultiplikacja

- Identyfikacja aplikacji komunikujących się poprzez numery portów.
- **Przykład**: Port 80 dla HTTP, port 443 dla HTTPS.

2. Podział i scalanie danych

- Segmentacja dużych porcji danych na mniejsze części (segmenty).
- Ponowne składanie danych w całość po dotarciu do odbiorcy.

3. Kontrola niezawodności przesyłu danych

- Oferowanie mechanizmów retransmisji utraconych pakietów.
- **Protokół**: TCP.

4. Kontrola przepływu

- Dostosowywanie tempa przesyłu danych między nadawcą a odbiorcą, aby uniknąć przepełnienia buforów.
- **Mechanizm**: Sliding Window w TCP.

5. Zapewnienie integralności danych

- Weryfikacja poprawności danych za pomocą sum kontrolnych.
- **Protokół**: TCP, UDP.

6. Opcjonalna obsługa strumieni i wielu kanałów

- SCTP umożliwia przesyłanie wielu strumieni danych w jednym połączeniu.

Adresacja typu broadcast i multicast w sieci IP

Broadcast – tylko UDP i IPv4:

- Wyszukiwanie zasobów – wyszukiwanie połączenia/usługi bez znajomości adresu
- ARP (Address Resolution Protocol)
- BOOTP (Bootstrap Protocol)
- NTP (Network Time Protocol)
- Demony wyznaczania tras (routed/RIPv1)

Adresy rozgłaszania

- **Adres rozgłaszania skierowany do podsieci**:
  - Np. 149.156.114.**255** dla sieci o masce 24
- **Adres ograniczonego rozgłaszania**:
  - 255.255.255.255
- Rutery nie powinny przekazywać datagramów rozgłoszeniowych
- Gniazdo UDP musi mieć ustawioną opcję SO_BROADCAST

Multicast – IPv6, opcja w IPv4, tylko UDP:

Do odebrania pakietu multicast stacja musi przyłączyć się do grupy multicastowej.

- Adresy **klasy D** od **224.0.0.0** do **239.255.255.255**
- Adresy **224.0.0.0/24** – zakres lokalny dla łącza do wykrywania topologii sieci i zarządzania, nierutowalne
- Dla IPv6 – **ff00::/8**
- Odwzorowanie w ramki Ethernet na adres grupowy zarządzany globalnie
  - Dla IPv4 – **01:00:5e**:
  - Dla IPv6 – **33:33:**

Bezpieczeństwo infrastruktury krytycznej i systemów sterowania przemysłowego

Niezawodność i cyberbezpieczeństwo infrastruktury krytycznej

Infrastruktura krytyczna – obiekty budowlane, urządzenia, instalacje, usługi kluczowe dla bezpieczeństwa państwa i jego obywateli oraz zapewnieniu sprawnego funkcjonowania administracji publicznej, a także instytucji i przedsiębiorców

Obejmuje to systemy:

- zaopatrzenia w energię, surowce energetyczne i paliwa,
- łączności,
- sieci teleinformatycznych,
- finansowe,
- zaopatrzenia w żywność,
- zaopatrzenia w wodę,
- ochrony zdrowia,
- transportowe,
- ratownicze,
- zapewniające ciągłość działania administracji publicznej,
- produkcji, składowania, przechowywania i stosowania substancji chemicznych i promieniotwórczych, w tym rurociągi substancji niebezpiecznych.

Zarządzanie Ciągłością Działania – np. monitorowanie SCADA, utrzymanie monitoringu i alarmów, w przypadku awarii dostępność chociaż centrum obsługi klienta (chroni reputacje, zwiększa odporność organizacji, zmniejsza ryzyko)

Niezawodność, dostępność, wytrzymałość, odporność, tolerancja na błędy

HAZOP – Analiza Zagrożeń i Operatywności (Hazard and Operability Study) to szczegółowa metoda analizy ryzyka służąca do identyfikacji potencjalnych zagrożeń oraz problemów w procesach operacyjnych.

- Polega na słowach kluczowych – czyli odchylenia od normy są opisywane jako: "więcej niż", "brak", "mniej niż" itp.

FMEA – (Failure Mode and Effects Analysis) jest systematyczną metodą analizy zaprojektowaną, by zidentyfikować potencjalne tryby awarii produktu lub procesu, ocenić ryzyko związane z tymi awariami i wskazać działania mające na celu eliminację lub zmniejszenie prawdopodobieństwa ich wystąpienia.

- Chodzi o znalezienie działań zapobiegawczych. Sprawdza prawdopodobieństwo wystąpienia awarii, jej skutków i możliwości wykrycia.

Standardy i normy:

- **ISO 22301** – zarządzanie ciągłością działania (systemy i wymagania),
- **ISO 27001** – zarządzanie bezpieczeństwem informacji,

- **NIST SP 800–34** – planowanie ciągłości działania i odzyskiwania po awarii IT,
- **IEC 62443** – bezpieczeństwo cybernetyczne w przemyśle.

Techniki analizy zagrożeń i analizy ryzyka

Analiza ryzyka pozwala organizacjom nie tylko zrozumieć ryzyka i zagrożenia, lecz także odkryć, jak mogą one oddziaływać na siebie nawzajem i na całość działalności organizacji.

Analiza FMEA (Failure Mode and Effects Analysis) tłumaczone często jako analiza rodzajów i skutków możliwych błędów) to metodyka, która pozwala identyfikować, jakie procesy lub produkty mogą zawieść, oraz oceniać wpływ tych awarii.

Analiza SWOT – (Strengths Weaknesses Opportunities Threats) – umożliwia określenie pozycji firmy na chwilę obecną oraz działań na przyszłość. Może dotyczyć konkretnego projektu.

STRIDE to metoda modelowania zagrożeń opracowana przez Microsoft, która pomaga identyfikować potencjalne zagrożenia bezpieczeństwa w systemach komputerowych:

- **S – spoofing** (naruszone: authentication) – udawanie kogoś innego
- **T – tampering** (naruszone: integrity) – modyfikacja danych/pamięci itp.
- **R – repudiation** (naruszone: non–repudiation) – zaprzeczenie wykonania akcji
- **I – information disclosure** (naruszone: confidentiality) – udostępnienie komuś informacji
- **D – denial of service** (naruszone: availability) – przeciążenie, zajęcie zasobów –> coś nie działa, jak powinno
- **E – elevation of privilege** (naruszone: authorization) – uzyskanie dostępu do nieprzeznaczonych dla nas danych/systemów itp.

Analiza scenariuszy polega na rozważaniu różnych potencjalnych przyszłych zdarzeń (scenariuszy) i ocenie ich wpływu na projekt lub organizację.

Wykres Ishikawy to wizualne narzędzie służące do identyfikacji, śledzenia i analizowania potencjalnych przyczyn konkretnego problemu w procesie lub systemie. (diagram rybiej ości)

- Promuje holistyczne podejście do problemu – zamiast skupiać się na pojedynczej przyczynie

Standardowy wykres Ishikawy (6M): **metody**, **maszyny**, **ludzie („manpower”)**, **materiały**, **środowisko („mother nature”)** i **pomiar („measurement”)**.

- Dla 8M jeszcze **zarządzanie („management”)** i **utrzymanie („maintenance”)**.
- W przeciwieństwie do szacowania ryzyka analiza ryzyka ma szerszy zakres, bo próbuje ogarnąć jakie ryzyka istnieją – potem dopiero można je skategoryzować zgodnie z szacowaniem

Oceny ryzyka:

- **ad–hoc**: przeprowadzane spontanicznie – testują szybkie reagowanie
- **dedukcyjne**: opierają się na logicznym wnioskowaniu od ogółu do szczegółu, gdzie wyniki zależą od znanych zasad
- **indukcyjne**: od szczegółu do ogółu, wykorzystując dane i obserwacje do formułowania ogólnych wniosków
- **ukierunkowane**: skoncentrowane na konkretnym zagrożeniu

Polityka cyberbezpieczeństwa w przedsiębiorstwie

Polityka cyberbezpieczeństwa w przedsiębiorstwie to zbiór zasad, procedur i wytycznych mających na celu ochronę systemów informacyjnych, danych oraz zasobów przed zagrożeniami i atakami w cyberprzestrzeni.

Celem polityki jest zapewnienie **integralności**, **poufności**, **dostępności** i **ciągłości działania systemów informacyjnych**.

Kluczowe elementy polityki cyberbezpieczeństwa:

- Określenie celów i zakresu polityki
- Zarządzanie dostępem
- Ochrona danych
- Zarządzanie ryzykiem
- Monitorowanie i audyt
- Zarządzanie incydentami bezpieczeństwa
- Szkolenie i świadomość pracowników
- Zarządzanie urządzeniami mobilnymi i BYOD (Bring Your Own Device)
- Zgodność z przepisami i standardami

Korzyści z wdrożenia polityki cyberbezpieczeństwa:

- Ochrona danych wrażliwych
- Zwiększenie odporności na ataki
- Zgodność z regulacjami
- Zwiększenie świadomości pracowników
- Zarządzanie ryzykiem

Praktyczne wykorzystanie informacji z otwartych źródeł

Biały wywiad

Biały wywiad (OSINT) to metoda pozyskiwania informacji z ogólnodostępnych źródeł, takich jak Internet, media tradycyjne (prasa, telewizja, radio), publikacje naukowe, rejestry publiczne, fora internetowe czy media społecznościowe. Jest to legalna i etyczna forma wywiadu, o ile korzystanie z informacji odbywa się zgodnie z obowiązującym prawem.

Zastosowania:

- Analiza danych wyciekających do publicznej przestrzeni,
- służby wywiadowcze wykorzystują to do monitorowania potencjalnych zagrożeń czy analizy informacji o grupach przestępczych,

- wywiady konkurencyjne, analiza rynkowa, identyfikacja trendów

Narzędzia OSINT

1) **Wyszukiwarki internetowe** (Google Dorking)
2) **Narzędzia do analizy mediów społecznościowych**
3) **Skanery sieciowe** (np. Maltego, Shodan, Spiderfoot)
4) **Bazy danych** i **rejestry publiczne** (KRS (Krajowy Rejestr Sądowy), rejestry handlowe, rejestry patentowe, Whois)
5) **Boty do scrapowania danych** ze stron internetowych
6) **Multimedia** (metadane EXIF (lokalizacja, analiza wizualna) – exiftool)

Pozyskiwanie danych i tworzenie raportów

Etapy pozyskiwania danych i tworzenia raportów OSINT:

1) **Planowanie i określanie celu**: stworzenie planu działania i listy potencjalnych źródeł informacji (cele to np. analiza konkurencji czy śledztwo kryminalne)
2) **Pozyskiwanie danych**: zbieranie danych za pomocą narzędzi wymienionych powyżej. Przykłady narzędzi:
    - **Shodan** – analiza urządzeń sieciowych
    - **Maltego** – wizualizacja powiązań między danymi, łączy wiele narzędzi OSINT
    - **Recon–ng** – zbieranie informacji o domenach, osobach, firmach
3) **Analiza danych**: filtrowanie, eliminacja duplikatów i informacji nieistotnych, weryfikacja wiarygodności, łączenie informacji, szeroka analiza
4) **Tworzenie raportu**: przedstawienie wyników analizy w sposób zrozumiały i użyteczny dla odbiorcy. Raport powinien zawierać:
    - **Streszczenie**: opis celu, metod i najważniejszych wniosków
    - **zastosowaną metodykę**: użyte narzędzia i źródła
    - **Wyniki** analizy: zebranie wszystkich danych, wniosków i powiązań. Tutaj dodaje się wykresy, mapy, diagramy itp.
    - **Rekomendacje**: konkretne kroki do podjęcia na podstawie wyników analizy.
    - **Załączniki**: dodatkowe dane, surowe wyniki, źródła